

Graph Embeddings Motivated by Greedy Routing

zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Roman Prutkin

aus Ekaterinburg

Tag der mündlichen Prüfung:	29. November 2017
Erster Gutachter:	Prof. Dr. Dorothea Wagner
Zweiter Gutachter:	Prof. Dr. Martin Nöllenburg
Dritter Gutachter:	Prof. Dr. Michael Kaufmann

To my family

DANKSAGUNG

Als Erstes möchte ich mich bei Dorothea Wagner bedanken für die Möglichkeit, an ihrem Lehrstuhl meiner Promotion nachzugehen. Ich weiß es sehr zu schätzen, dass ich ohne Zeitdruck und unter äußerst komfortablen Rahmenbedingungen Forschungsfragen untersuchen konnte, die mich interessieren. Bei der Wahl der Forschungsthemen war ich immer frei und bekam von Dorothea stets Rat und Unterstützung.

Michael Kaufmann danke ich, dass er sich bereit erklärt hat, meine Dissertation zu begutachten, sowie für seine Flexibilität bei der Terminfindung für die Promotionsprüfung.

Martin Nöllenburg und Ignaz Rutter haben mich sowohl bei meiner Diplomarbeit als auch während meiner Promotion hervorragend betreut. Ich bin sehr froh, dass ich von eurem Fachwissen und Erfahrung profitieren und von euch viel lernen konnte. Die Zusammenarbeit mit euch hat mir Spaß gemacht.

Ich bedanke mich bei meinen Koautoren Therese Biedl, Thomas Bläsius, Xiaojj Chen, Carsten Dachsbacher, Fabian Fuchs, Alexander E. Holroyd, Anton Kaplanyan, Boris Klemz, Bongshin Lee, Moritz von Looz, Henning Meyerhenke, Lev Nachmanson, Benjamin Niedermann, Martin Nöllenburg, Nathalie Henry Riche und Ignaz Rutter.

Bei meinen aktuellen und ehemaligen Kollegen Lukas Barth, Moritz Baum, Thomas Bläsius, Guido Brückner, Valentin Buchhold, Julian Dibbelt, Fabian Fuchs, Andreas Gemsa, Sascha Gritzbach, Michael Hamann, Tanja Hartmann, Andrea Kappes, Tamara Mchedlidze, Benjamin Niedermann, Martin Nöllenburg, Thomas Pajor, Marcel Radermacher, Ignaz Rutter, Ben Strasser, Markus Völker, Franziska Wegner, Matthias Wolf, Tim Zeitz und Tobias Zündorf bedanke ich mich für die angenehme und lockere Atmosphäre am Institut und die vielen Tischfußballspiele, Kino-, Bastel- und Spieleabende, Wanderungen und Fußball-Sessions im Soccer Center. Besonderer Dank geht an Fabian Fuchs, der mit mir das Sensornetze-Büro geteilt hat.

Ich danke meiner Frau Anna, meinen Eltern Olga und Ilya und dem Rest meiner Familie für eure Inspiration und grenzenlose Unterstützung während meiner Zeit als Doktorand, insbesondere in der Zeit des Schreibens dieser Arbeit.

DEUTSCHE ZUSAMMENFASSUNG

In dieser Arbeit untersuche ich Probleme aus Algorithmischer Geometrie und Graphentheorie, die *Greedy Routing* betreffen. Ich konzentriere mich insbesondere auf Greedy Routing in geometrisch eingebetteten Graphen, welches wie folgt definiert ist. Gegeben sei ein Graph $G = (V, E)$, dessen Knoten V Koordinaten zugeordnet wurden, beispielsweise Punkte in der euklidischen Ebene. Kanten E symbolisieren die Möglichkeit der direkten bidirektionalen Kommunikation zwischen Knoten. Jeder Knoten kennt seine eigenen Koordinaten und die seiner direkten Nachbarn in G . Beim Routing von Nachrichten in diesem Netzwerk nehmen wir zusätzlich an, dass jede Nachricht die Koordinaten des Ziels enthält.

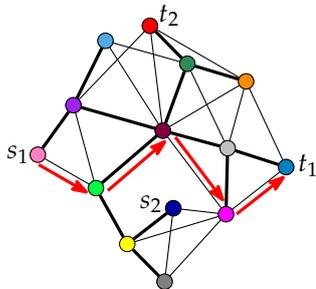


Abbildung 1: Greedy Routing ist erfolgreich zwischen dem Start s_1 und Ziel t_1 (roter Pfad). Knoten s_2 ist ein lokales Minimum für das Ziel t_2 .

Das Grundproblem von Greedy Routing ist, dass Nachrichten in lokalen Minima stecken bleiben können, wo kein Nachbarknoten näher am Ziel liegt (siehe zum Beispiel Abbildung 1 für den Startknoten s_2 und Zielknoten t_2).

Für einen gegebenen Graphen bestimmt die Wahl der Knotenkoordinaten die Erfolgsrate von Greedy Routing. Für den Graphen in Abbildung 1 ist eine andere Koordinatenzuweisung in Abbildung 2 dargestellt. In der letzteren Einbettung ist Greedy Routing von jedem Start- zu jedem Zielknoten erfolgreich. Grapheneinbettungen mit dieser Eigenschaft werden *Greedy-Einbettungen* bzw. *Greedy-Zeichnungen* genannt. Die Untersuchung

Unter den obigen Annahmen ist die folgende einfache Routing-Strategie als Greedy Routing bekannt. Für eine eingehende Nachricht berechnet jeder Knoten die euklidischen Distanzen zum Ziel ausgehend von sich selbst und von jedem seiner Nachbarn und gibt anschließend die Nachricht an einen Nachbarn weiter, der näher am Ziel liegt als der Knoten selbst. Abbildung 1 zeigt einen geometrisch eingebetteten Graphen sowie einen möglichen Pfad beim Greedy Routing zwischen dem Startknoten s_1 und Ziel t_1 . Greedy Routing ist beispielsweise einer der beiden Routing-Modi im Protokoll *GPSR (Greedy Perimeter Stateless Routing)* für drahtlose Sensornetze.

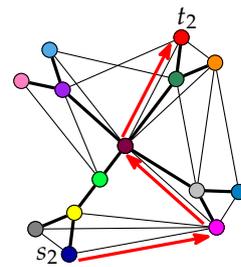


Abbildung 2: Eine Einbettung des Graphen aus Abbildung 1 mit einer anderen Zuweisung der Knotenkoordinaten. Identische Knoten haben die gleiche Farbe in beiden Abbildungen. Greedy Routing ist nun immer erfolgreich.

der Greedy-Einbettungen wird in der Literatur durch Routing in drahtlosen Sensornetzwerken motiviert.

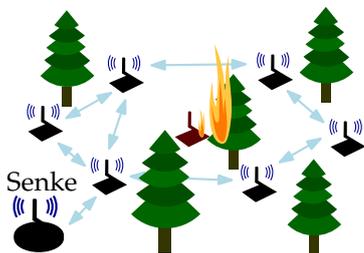


Abbildung 3: Sensornetze werden zur Erkennung von Waldbränden verwendet.

Drahtlose Sensornetze bzw. Sensornetze (engl. *wireless sensor networks*) sind Netzwerke von kleinen mit Sensoren ausgestatteten Rechenknoten. Die Knoten sind räumlich verteilt und können untereinander drahtlos kommunizieren. Obwohl einzelne Knoten typischerweise nur über begrenzte Rechenkapazitäten sowie begrenzte Batterien verfügen, können die Knoten ein Netzwerk bilden und eine Aufgabe in Zusammenarbeit erfüllen. Sie können beispielsweise Temperatur, Feuchtigkeit, Konzentration von Kohlenmonoxid in der Luft usw. überwachen und diese Daten an eine Basisstation weiterleiten als Teil eines Systems, das Waldbrände erkennt und überwacht (siehe Abbildung 3). Ein solches Netzwerk kann seine Aufgabe weiterführen, auch wenn einige Knoten ausfallen. Anwendungsgebiete von Sensornetzen sind Militär, Umwelt, Gesundheitswesen und Sicherheit.

Die Vision, wie man Greedy-Einbettungen für das Routing in Sensornetzen anwenden könnte, wird in der Literatur wie folgt geschildert. Das Sensornetz berechnet eine Greedy-Einbettung von seinem Kommunikationsgraphen und teilt jedem Knoten seine eigenen Koordinaten in dieser Einbettung mit (die sogenannten *virtuellen Koordinaten*) sowie die virtuellen Koordinaten der Nachbarknoten. Enthält nun jede Nachricht die virtuellen Koordinaten des Zielknotens, kann jeder Knoten die Kenntnis seiner virtuellen Koordinaten und der seiner Nachbarn nutzen, um die Nachricht mittels Greedy Routing weiterzuleiten. Da die virtuellen Koordinaten aus einer Greedy-Einbettung stammen, ist Greedy Routing nun immer erfolgreich. In dieser Arbeit untersuche ich die Realisierbarkeit dieser Vision aus dem Blickwinkel der Graphentheorie und erhalte neue Erkenntnisse über die Frage, welche Graphen eine Greedy-Einbettung zulassen.

Routing in Sensornetzen ist nicht die einzige Motivation für die Untersuchung von Greedy-Einbettungen von Graphen und anderen verwandten Einbettungsarten. Kriterien wie etwa möglichst wenige lokale Minima spielen eine Rolle, wenn eine Netzwerkzeichnung Nutzern helfen soll, Pfade im Netzwerk zu finden. Dazu wurde in den letzten Jahren eine Reihe von verschiedenen Zeichnungskonventionen vorgeschlagen, nämlich die bereits erwähnten Greedy-Zeichnungen sowie (*stark*) *monotone*, *Self-Approaching*- und *Increasing-Chord*-Zeichnungen (siehe Abbildung 4). Ich fasse das Problem, eine für die Pfadsuche geeignete Netzwerkeinbettung zu konstruieren, wie folgt auf: Finde für einen gegebenen Graphen Knotenkoordinaten in \mathbb{R}^2 , die verwendet werden können

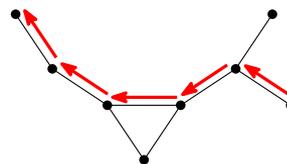


Abbildung 4: Eine Increasing-Chord-Graphzeichnung. Für jeden Start- und Zielknoten existiert immer ein Pfad, entlang dessen Kanten die Distanz zum Ziel kontinuierlich abnimmt.

Routing in Sensornetzen ist nicht die einzige Motivation für die Untersuchung von Greedy-Einbettungen von Graphen und anderen verwandten Einbettungsarten. Kriterien wie etwa möglichst wenige lokale Minima spielen eine Rolle, wenn eine Netzwerkzeichnung Nutzern helfen soll, Pfade im Netzwerk zu finden. Dazu wurde in den letzten Jahren eine Reihe von verschiedenen Zeichnungskonventionen vorgeschlagen, nämlich die bereits erwähnten Greedy-Zeichnungen sowie (*stark*) *monotone*, *Self-Approaching*- und *Increasing-Chord*-Zeichnungen (siehe Abbildung 4). Ich fasse das Problem, eine für die Pfadsuche geeignete Netzwerkeinbettung zu konstruieren, wie folgt auf: Finde für einen gegebenen Graphen Knotenkoordinaten in \mathbb{R}^2 , die verwendet werden können

nen, um auf dem Graphen mit lokalen Entscheidungen zu routen, und sodass man Pfade finden kann, die immer Fortschritte in Richtung ihrer Ziele machen. Dies ist das zentrale Problem, das in dieser Arbeit untersucht wird.

ÜBERBLICK UND BEITRAG

Ich betrachte mehrere Arten von Graphzeichnungen, die durch Greedy Routing auf geometrisch eingebetteten Graphen motiviert sind. Das zentrale Problem, das ich untersucht habe, ist, zu verstehen, welche Graphen eine Greedy-, Self-Approaching- oder Increasing-Chord-Zeichnung zulassen. Meine Arbeit erweitert den aktuellen Kenntnisstand zu dieser Frage um neue Erkenntnisse. Auf dem Weg zu einer vollständigen Charakterisierung von Graphen, die solche Zeichnungen zulassen, konzentriere ich mich auf gängige und wichtige Graphklassen wie Bäume, Triangulierungen und dreifach-zusammenhängende planare Graphen, die in diesem Forschungsbereich häufig betrachtet werden.

Außerdem untersuche ich die Komplexität des Problems, Polygone und Graphzeichnungen in Teilbereiche zu zerlegen, die Greedy Routing unterstützen (siehe Abbildung 5). Dieses Zerlegungsproblem entstammt direkt aus einem für drahtlose Sensornetzwerke vorgeschlagenen Routing-Algorithmus (Tan und Kermarrec, *IEEE/ACM Trans. Networking* 20.3 (2012), 864–877) und ist stark verbunden mit Increasing-Chord-Zeichnungen.

Euklidische Greedy-Zeichnungen von Bäumen

Im Zusammenhang mit dem Einbetten von Graphen in \mathbb{R}^2 , um Greedy Routing zu unterstützen, ist folgendes Problem der „Heilige Gral“: Charakterisiere die Graphen, die eine Greedy-Zeichnung in \mathbb{R}^2 besitzen. Dieses Problem zog großes Interesse der Graph-Drawing-Gemeinschaft auf sich. Obwohl die Existenz von Greedy-Zeichnungen für mehrere Graphklassen gezeigt werden konnte, bleibt eine vollständige Charakterisierung von Graphen, die eine Greedy-Zeichnung in \mathbb{R}^2 haben, ein bislang unerreichtes Ziel. Überraschenderweise blieb das Problem für eine solch natürliche Graphklasse wie Bäume offen. In dieser Arbeit charakterisiere ich alle Bäume, die eine Greedy-Zeichnung in \mathbb{R}^2 besitzen. Dies beantwortet eine Frage von Angelini et al. (*Networks* 59.3 (2012), 267–274) und ist ein wichtiger Schritt in Richtung einer Charakterisierung der Greedy-einbettbaren Graphen.

Über Self-Approaching- und Increasing-Chord-Zeichnungen von dreifach-zusammenhängenden planaren Graphen

Ich untersuche Self-Approaching- und Increasing-Chord-Zeichnungen für zwei gängige Graphklassen: Triangulierungen und dreifach-zusammenhängende planare Graphen. Ich zeige, dass in \mathbb{R}^2 alle Triangulierungen Increasing-Chord-Zeichnungen besitzen und dass für *planare 3-Bäume* Planarität sichergestellt werden kann. Außerdem beweise ich, dass *binäre Kakteen*, eine Graphklasse, die für die

Konstruktion von Greedy-Zeichnungen von dreifach-zusammenhängenden planaren Graphen entscheidend war, nicht immer Self-Approaching-Zeichnungen haben. Ich zeige, dass stark monotone (und damit Increasing-Chord-) Zeichnungen von Bäumen und binären Kakteen in manchen Fällen eine exponentielle Auflösung benötigen, und beantworte dadurch eine offene Frage von Kindermann et al. (*Graph Drawing*, 2014, 488–500). Ich beweise, dass das Gleiche für Greedy-Zeichnungen von binären Kakteen gilt. Ich zeige, dass dreifach-zusammenhängende planare Graphen Increasing-Chord-Zeichnungen in der hyperbolischen Ebene besitzen, und charakterisiere Bäume, die solche Zeichnungen haben.

Zerlegung von Graphzeichnungen und triangulierten einfachen Polygonen in Greedy-Routbare Regionen

Als nächstes betrachte ich Greedy Routing in kontinuierlichen Domänen und entdecke einen starken Zusammenhang zu Self-Approaching- und Increasing-Chord-Zeichnungen von Graphen. Mehrere vorgeschlagene Ansätze für das Routing in drahtlosen Sensornetzwerken basieren auf der Idee, das Netzwerk in Komponenten zu zerlegen, sodass in jeder von ihnen Greedy Routing mit großer Wahrscheinlichkeit erfolgreich ist. Eine globale Datenstruktur von vorzugsweise kleiner Größe speichert die Interkonnektivität zwischen Komponenten.

Eine *Greedy-Routbare Region (GRR)* ist eine abgeschlossene Teilmenge von \mathbb{R}^2 , in der jeder beliebige Zielpunkt von jedem beliebigen Startpunkt aus mit Greedy Routing erreicht werden kann (siehe Abbildung 5).

Tan und Kermarrec (*IEEE/ACM Trans. Networking* 20.3 (2012), 864–877) schlugen einen Routing-Algorithmus vor, der darauf basiert, den Netzwerkbereich in wenige GRRs zu partitionieren. Sie zeigten, dass es NP-schwer ist, polygonale Regionen mit Löchern minimal zu zerlegen.

Ich untersuche minimale GRR-Zerlegung für planare geradlinige Zeichnungen von Graphen, was eine natürliche Anpassung des GRR-Zerlegungsproblems für Polygone darstellt. In diesem Kontext stimmen die GRRs mit Increasing-Chord-Zeichnungen von Bäumen überein. Ich zeige, dass die minimale Zerlegung immer noch NP-schwer für Graphen mit Zyklen und sogar für Bäume ist, aber für Bäume in Polynomialzeit optimal gelöst werden kann, wenn nur bestimmte Arten von GRR-Kontakten zugelassen sind (z.B., wenn sich GRRs nicht kreuzen dürfen). Darüber hinaus gebe ich eine 2-Approximation für löcherfreie Polygone an für den Fall, wenn eine gegebene Triangulierung eingehalten werden muss.



Abbildung 5: Sei die Knotendichte innerhalb der Netzwerkgrenze (grau) nah an unendlich. Greedy Routing leitet eine Nachricht entlang der geraden Linie zum Ziel weiter. Wird die Grenze getroffen, gleitet die Nachricht entlang der Grenze, solange es die Distanz zum Ziel verringert. Dies ist erfolgreich beim Routing von Punkt s_1 zu t_1 (gestrichelte Bahn). Greedy Routing von Punkt s_2 zu t_2 bleibt in einem lokalen Minimum p stecken. Greedy Routing innerhalb jeder der beiden Partitionen (hell- und dunkelgrau) ist immer erfolgreich.

CONTENTS

1	INTRODUCTION	1
2	RELATED WORK	7
2.1	Routing in wireless ad hoc and sensor networks	7
2.1.1	Geographic routing	8
2.2	Greedy embeddings	12
2.2.1	Graphs admitting Euclidean greedy embeddings	12
2.2.2	Non-Euclidean greedy embeddings	13
2.2.3	Succinctness	13
2.3	Graph drawings with geodesic-path tendency	14
2.3.1	Monotone drawings	14
2.3.2	Geometric spanners	16
2.3.3	Self-approaching and increasing-chord drawings	16
2.4	Greedy geometric routing in continuous domains	19
2.4.1	Beacon-based routing	19
2.4.2	Network decomposition for routing	20
3	PRELIMINARIES	23
3.1	Graphs, paths and connectivity	23
3.2	Graph drawings and polygons	24
3.2.1	Greedy, monotone, self-approaching and increasing-chord drawings	25
3.2.2	Used geometric notation	26
I	GRAPH EMBEDDINGS MOTIVATED BY GREEDY ROUTING: EXISTENCE	27
4	EUCLIDEAN GREEDY DRAWINGS OF TREES	29
4.1	Introduction	29
4.1.1	Contribution	29
4.2	Preliminaries	30
4.2.1	Example: independent stars	35
4.2.2	Outline of the characterization	36
4.2.3	Shrinking lemma	36
4.3	Opening angles of rooted trees	39
4.4	Arranging rooted subtrees with open angles	45
4.5	Characterizing greedy-drawable binary trees	53
4.6	Recognition algorithm	56
4.6.1	Maximum degree 4	56
4.6.2	Maximum degree 5 and above	58
4.7	Conclusion	59
5	ON SELF-APPROACHING AND INCREASING-CHORD DRAWINGS OF 3-CONNECTED PLANAR GRAPHS	61

5.1	Introduction	61
5.1.1	Contribution	62
5.2	Preliminaries	62
5.3	Graphs with self-approaching drawings	65
5.3.1	Increasing-chord drawings of triangulations	65
5.3.2	Exponential worst case resolution	66
5.3.3	Non-triangulated cactuses	73
5.4	Planar increasing-chord drawings of 3-trees	79
5.5	Self-approaching drawings in the hyperbolic plane	81
5.6	Bounded dilation for Euclidean greedy drawings of cactuses	84
5.7	Conclusion	86
II	GREEDY ROUTING IN CONTINUOUS DOMAINS	89
6	PARTITIONING INTO GREEDILY ROUTABLE REGIONS	91
6.1	Introduction	91
6.1.1	Contribution	92
6.2	Preliminaries	93
6.2.1	Greedly Routable Regions	93
6.2.2	Splitting graph drawings at non-vertices	97
6.2.3	Types of GRR contacts in plane straight-line graph drawings	98
6.3	NP-completeness for graphs with cycles	99
6.4	Trees	107
6.4.1	NP-completeness	108
6.4.2	Polynomial-time algorithms for restricted types of contacts	112
6.5	Triangulations	124
6.6	Heuristics for simple polygons	126
6.7	Conclusion	127
7	CONCLUSION	133
III	APPENDIX	137
A	APPENDIX	139
	BIBLIOGRAPHY	143
	LIST OF PUBLICATIONS	157

INTRODUCTION

In this thesis, I study problems from computational geometry and graph theory concerned with *greedy routing*. In particular, I focus on greedy routing on geometrically embedded graphs, which is defined as follows. We are given a graph $G = (V, E)$ whose nodes V have been assigned coordinates that are points in a metric space, e.g., the Euclidean plane. Edges E denote the possibility of direct bidirectional communication between nodes. Every node knows its own coordinates and those of its immediate neighbors in G . For routing between pairs of nodes in this network, we additionally assume that every routed message contains the coordinates of its destination.

Under the above assumptions, the following simple routing strategy is known as *greedy routing* or *greedy forwarding*. For an incoming message, a node computes the Euclidean distances from itself and from every neighbor to the destination and then simply passes the message to a neighbor that is closer to the destination than the node itself. Figure 6a shows a geometrically embedded graph and a possible path when routing from node s_1 to the destination t_1 greedily. For exam-

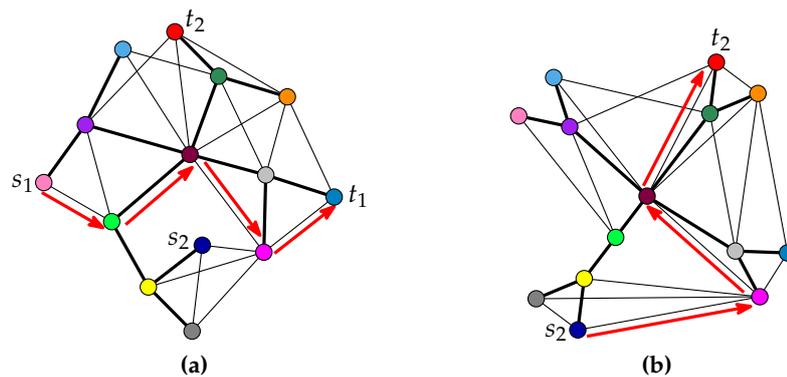


Figure 6: Greedy routing on a geometrically embedded graph. In embedding (a), greedy routing is successful from node s_1 to node t_1 ; see the red path. However, node s_2 is a local minimum for the destination t_2 . (b) An embedding of the graph from (a) with a different assignment of node coordinates. Identical nodes have the same color in (a) and (b). Greedy routing is always successful on the new coordinates. For source s_2 and destination t_2 , a possible path for greedy routing is depicted.

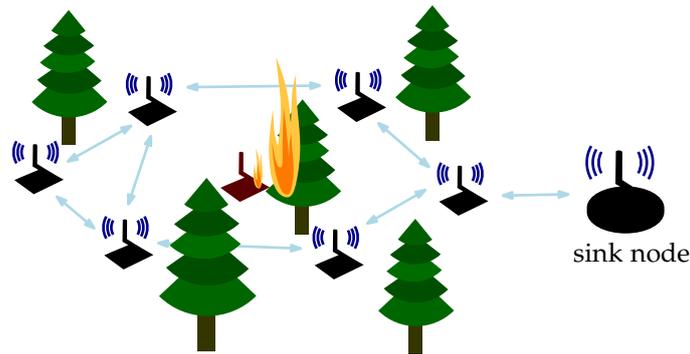


Figure 7: Wireless sensor networks are used to detect forest fires.

ple, greedy forwarding is one of the two routing modes in the *Greedy Perimeter Stateless Routing* protocol for wireless sensor networks [KK00; Kim+05a].

The basic problem of greedy routing is that messages can get stuck at local minima, or *voids*, where no node closer to the destination exists; see node s_2 in Figure 6a for the destination t_2 .

For a given graph, the choice of node coordinates determines the success rate of greedy routing. For example, for the graph in Figure 6a, consider a different coordinate assignment shown in Figure 6b. In this graph embedding, greedy routing is successful for every pair of source and destination nodes. Graph embeddings with this property are called *greedy embeddings* or *greedy drawings*. Equivalently, every pair of vertices in a greedy embedding is connected by a *distance-decreasing path*, i.e., a path on which every vertex v is closer to the path's destination than all vertices preceding v on the path. The study of greedy drawings is motivated in the literature by routing in *wireless ad hoc* and *sensor networks* [Rao+03; PR05; Kle07; Dha10; LM10; AFG10; EG11; WH14; DDF17].

Wireless sensor networks, or *sensornets*, are networks of small computing nodes equipped with sensors. The nodes are spatially distributed and can communicate wirelessly among each other. Although single nodes typically have limited computational capacities as well as limited batteries, the nodes can form a network and collaborate on a task, for example, monitor temperature, humidity, concentration of carbon monoxide in the air etc. and forward this data to a base station as part of a system that detects and monitors forest fires [DS05]; see Figure 7. Such a network can carry on its task even if some nodes are destroyed. Application areas of wireless sensor networks include military, environment, healthcare and security [Raw+14].

Applying greedy embeddings for routing in wireless sensor networks is envisioned in the literature as follows [Rao+03]. The sensornet computes a greedy embedding of its communication graph, and every network node is notified about its own coordinates in this embedding, the so-called *virtual coordinates*, as well as the virtual coordinates of the node's neighbors. Let every message contain the virtual coordinates of the destination node. Then, every node can use its knowledge of the virtual coordinates of itself, its neighbors and the destination to forward the

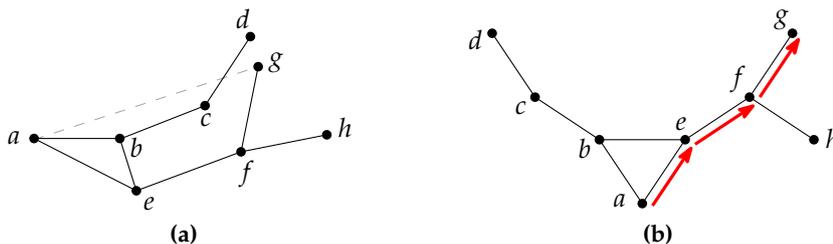


Figure 8: (a) When tracing a path from a to g , a user is likely to follow the path a - b - c - d first, before finding a solution. Example redrawn from [HEH09]. (b) An increasing-chord drawing of the same graph. For every pair of source and destination vertices, there is always an edge along which the distance towards the destination decreases continuously.

message greedily as described earlier. Greedy routing is now always successful, since virtual coordinates originate from a greedy embedding. The idea of greedy routing on virtual coordinates has inspired a number of routing algorithm proposals for sensor networks [Rao+03; NS03; Fan+05; Kle07; Sar+09; Sar+10]. In this thesis, I investigate the realizability of this vision from the graph-theoretic viewpoint and gain new insights into the question of which graphs admit a greedy embedding.

Routing in sensor networks is not the only motivation for studying greedy and related embeddings of graphs. Finding paths between two vertices is one of the most fundamental tasks users want to solve when considering network drawings [Lee+06]. Imagine yourself traveling in an unfamiliar city using public transportation. To find your way from station A to station B , you would typically use a map of the metro or tram network of the city and try to find a path from A to B on that map. Some drawings of a network are more suited for such path-finding tasks than others. One example are schematic drawings of metro or tram networks, which simplify line trajectories while accepting a distortion of geographic locations of stations.

Empirical studies have shown that when finding paths in a network drawing, users are more likely to follow edges that are directed towards the destination; see Figure 8a. This is known as *geodesic-path tendency* [HEH09; Pur+12]. Users perform better in path-finding tasks if following such edges lets them discover a path to the desired destination vertex [HEH09]. Over the last years a number of different drawing conventions implementing the notion of strong geodesic-path tendency have been suggested, namely the aforementioned *greedy drawings* [Rao+03], *(strongly) monotone drawings* [Ang+12] as well as *self-approaching* and *increasing-chord drawings* [Ala+13]. For example, Figure 8a shows an increasing-chord drawing of the graph in Figure 8b.

OVERVIEW AND CONTRIBUTION

In my thesis, I consider several graph drawing styles that are motivated by greedy routing on geometrically embedded graphs. The central problem I investigate is understanding which graphs have greedy, self-approaching and increasing-chord drawings and, in the positive case, constructing the actual drawings. On the path towards a complete characterization of graphs admitting such drawings, I focus on popular and important graph classes such as trees, triangulations and 3-connected planar graphs, as is common for this research area.

Furthermore, I study the complexity of partitioning graph drawings and polygons into a minimum number of components that support greedy routing; see Figure 9 for an example. This problem results directly from a routing algorithm proposed for sensornets [TK12] and is strongly related to increasing-chord graph drawings.

Chapter 4: Euclidean greedy drawings of trees

In the context of embedding graphs in \mathbb{R}^2 to support greedy routing, the following problem is the “holy grail”:

Problem. *Characterize graphs that admit a greedy drawing in \mathbb{R}^2 .*

This problem has attracted a lot of interest from the graph drawing and computational geometry communities; see Chapter 2 for an overview of the contributions. Although the existence of greedy embeddings has been shown for several graph classes, a complete characterization of graphs with a greedy drawing in \mathbb{R}^2 remains an elusive goal.

Surprisingly, the problem has been open for such a natural graph class as trees. In Chapter 4, I completely characterize the trees that admit a greedy embedding in \mathbb{R}^2 . This answers a question by Angelini et al. [ADF12] and is a further step in characterizing the graphs that admit Euclidean greedy embeddings.

Chapter 4 is based on joint work with Martin Nöllenburg [NP13; NP17].

Chapter 5: On Self-approaching and increasing-chord drawings of 3-connected planar graphs

An st -path in a drawing of a graph is *self-approaching* if during the traversal of the corresponding curve from s to any point t' on the curve the distance to t' is non-increasing. A path is *increasing-chord* if it is self-approaching in both directions. A drawing is self-approaching (increasing-chord) if any pair of vertices is connected by a self-approaching (increasing-chord) path. Self-approaching graph drawings are greedy drawings, but the converse does not hold in general. Due to stronger geodesic-path tendency, self-approaching and increasing-chord graph drawings are believed to be more suited to aid users at path-finding tasks than greedy drawings.

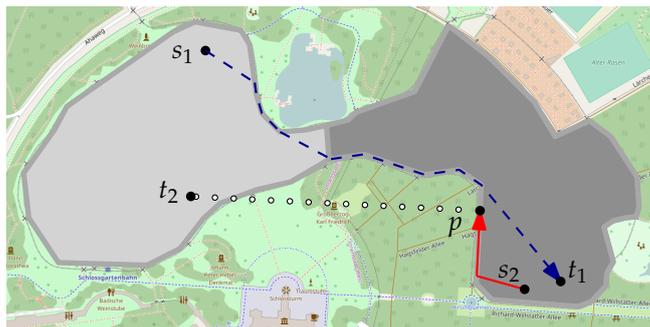


Figure 9: Inside a polygon, greedy routing will forward a message along the straight line towards its destination or, if the boundary is hit, the message will slide along the boundary, as long as it decreases the Euclidean distance to the destination. This is successful when routing from point s_1 to t_1 ; see the dashed trajectory. Greedy routing from point s_2 to t_2 gets stuck in a local minimum p . Greedy routing is always successful inside each of the two partitions (dark gray and light gray).

In Chapter 5, I study self-approaching and increasing-chord drawings of two popular graph classes, triangulations and 3-connected planar graphs. I show that in the Euclidean plane, triangulations admit increasing-chord drawings, and for *planar 3-trees* planarity can be ensured. Moreover, I show that *binary cactuses*, a graph class that has been crucial for constructing greedy drawings of 3-connected planar graphs, do not admit self-approaching drawings in general.

I prove that strongly monotone (and thus increasing-chord) drawings of trees and binary cactuses require exponential resolution in the worst case, answering an open question by Kindermann et al. [Kin+14]. Using the developed techniques, I show that the same holds for greedy drawings of binary cactuses, which proves a conjecture by Leighton and Moitra [MLo8, slide 79].

I show that 3-connected planar graphs admit increasing-chord drawings in the hyperbolic plane and characterize the trees that admit such drawings. Finally, I prove that Euclidean greedy drawings of trees and cactuses have bounded dilation.

Chapter 5 is based on joint work with Martin Nöllenburg and Ignaz Rutter [NPR14; NPR16].

Chapter 6: Partitioning graph drawings and triangulated simple polygons into greedily routable regions

In Chapter 6, I reveal strong connections of self-approaching and increasing-chord drawing styles to greedy routing in polygonal regions. Informally, when considering greedy drawings on one hand and routing in polygonal regions on the other, increasing-chord graph drawings can be viewed as an intermediate step between the two. This provides additional motivation for studying self-approaching and increasing-chord graph drawings.

Several proposed algorithms for routing in wireless sensor networks are based on decomposing the network into components such that in each of them greedy routing is likely to perform well [Fan+05; BGJ07; FM07; ZSG07; TBK09; ZSG09; Jia+15]. A global data structure of preferably small size is used to store interconnectivity between components. One such routing algorithm based on network decomposition has been proposed by Tan and Kermarrec [TK12]. In Chapter 6, I consider a polygon decomposition problem that arises in that algorithm.

A *greedily routable region* (GRR) is a closed subset of \mathbb{R}^2 , in which any destination point can be reached from any starting point by always moving in the direction with maximum reduction of the distance to the destination in each point of the path. The geographic routing approach proposed by Tan and Kermarrec [TK12] aims at dense wireless sensor networks with obstacles and is based on decomposing the network area into a small number of interior-disjoint GRRs. The authors showed that minimum decomposition is NP-hard for polygonal regions with holes and presented a simple heuristic, which does not offer an approximation guarantee. Figure 9 shows a minimum decomposition of a simple polygon in two GRRs.

I consider minimum GRR decomposition for plane straight-line drawings of graphs, which is a natural adjustment of the minimum GRR partition problem. Here, GRRs coincide with self-approaching drawings of trees. I show that minimum decomposition is still NP-hard for graphs with cycles and even for trees, but can be solved optimally for trees in polynomial time, if we allow only certain types of GRR contacts (e.g., we disallow GRRs to have proper intersections). Additionally, I give a 2-approximation for simple polygons, if a given triangulation has to be respected.

Chapter 6 is based on joint work with Martin Nöllenburg and Ignaz Rutter [NPR15; NPR17].

RELATED WORK

I start by giving a brief overview of routing algorithms for wireless ad hoc and sensor networks, with focus on greedy and geographic routing. A detailed survey of sensor network routing approaches is beyond the scope of this thesis; for this, I refer the reader to the books by Wagner and Wattenhofer [WW07], Boukerche [Bou08] and Akyildiz and Vuran [AV10] as well as the surveys by Al-Karaki and Kamal [AK04], Frey et al. [FRS09] and Pantazis et al. [PNV13].

2.1 ROUTING IN WIRELESS AD HOC AND SENSOR NETWORKS

The ability of a wireless sensor network to forward messages from one node to another, or point-to-point routing, is considered an important primitive [Fon+05]. Typically, a node can only communicate to a small subset of other nodes in its vicinity directly; we shall call such nodes *neighbors*. Therefore, a message may pass intermediate nodes before it reaches the destination node, i.e., the network must be able to perform *multi-hop* communication.

Numerous routing strategies for wireless ad hoc networks have been proposed in the literature. Routing protocols are commonly distinguished between *proactive* and *reactive* [Zolo7; FRS09]. Proactive protocols compute and maintain information about available paths in form of routing tables that are updated whenever the network topology changes. Due to the resulting significant communication and computation overhead, proactive approaches are considered to be not well suited for highly dynamic networks [FRS09]. Examples of proactive routing protocols are Optimized Link State Routing (OLSR) [RFC3626] and Destination Sequenced Distance Vector (DSDV) [PB94] protocols. Reactive approaches perform route discovery on demand. Examples of reactive routing protocols are Ad hoc On-demand Distance Vector (AODV) [PR99] and Dynamic Source Routing (DSR) [JM96].

When designing algorithms for wireless ad hoc and sensor networks, numerous parameters of the networks have to be taken into account, such as node density and distribution, transmission powers, signal attenuation, node mobility, etc. [Zolo7]. The resulting high number of degrees of freedom has led to a great number of proposed approaches as well as various proposals for their classification [AK04; Bou+08; FRS09; AM12].

2.1.1 Geographic routing

Routing algorithms in traditional IP-based networks use the global hierarchy of IP addresses [Com00]. For wireless sensor networks, building such global addressing schemes is considered challenging due to the potentially large number of sensor nodes [AK04; ZJ09]. A family of alternative routing and addressing strategies in wireless networks, known as *geographic* or *position-based* routing and addressing, uses node locations as addresses instead [KW05; GG12]. Geographic routing protocols are nearly stateless, since every node only needs to know the coordinates of itself, its immediate neighbors and of the current destination to make forwarding decisions [FRS09]. Node positions can be discovered using GPS or distance estimation based on signal strengths. Inquiry of destination position can be realized by a *location service* [Li+00].

2.1.1.1 Greedy routing

One simple geographic routing strategy is greedy routing. Upon receipt of a message, a node tries to forward it to a neighbor node that is closer to the destination than itself [Fin87; SL01]. For example, greedy routing is one of the two routing modes in the Greedy Perimeter Stateless Routing protocol (GPSR) [KK00]. Another local routing strategy is compass routing. It forwards the message to a neighbor, such that the direction from the node to this neighbor is closest to the direction from the node to the destination. Kranakis et al. [KSU99] showed that compass routing can produce loops even in plane triangulations. They also showed that compass routing is always successful on Delaunay triangulations. Bose et al. [Bos+02] showed that a combination of the two strategies, the *greedy-compass* algorithm, is successful on any triangulation. Neither greedy nor compass nor greedy-compass routing guarantee delivery in general.

When multiple neighbors reduce the distance to the destination during greedy routing, energy consumption and potential packet loss should be taken into account in practice. Forwarding to the neighbor closest to the destination might result in using long links with higher loss probability. Therefore, a balance between long but lossy and short but reliable links must be found. Seada et al. [Sea+04] use a local metric that is the product of distance improvement and packet reception rate. To reduce loss probability for long links, one might consider increasing the transmission power on demand. Li et al. [Li+05] proposed another local routing metric for energy-efficient greedy routing with adjustable transmission powers.

A strategy similar in spirit to greedy routing is *geographic opportunistic routing* [ZR03; Zen+07; Cha15]. Here, nodes do not store the coordinates of their neighbors. Instead, upon receipt of a message, a node broadcasts it to all neighbors together with the node's own coordinates. Those neighbors that receive the message and are closer to the destination compete with each other and finally agree on which of them shall retransmit the message.

2.1.1.2 Greedy routing with recovery

To overcome the problem of local minima, geographic routing algorithms often employ two alternating modes [KK00; Bos+01; LLM06; KWZ08a; KWZ08b]. In the *greedy* mode, every node considers the locations of its neighbor nodes and tries to forward the message to a neighbor that advances the message towards the destination. This advance is usually defined in terms of decreasing the Euclidean distance towards the destination [GG12], which corresponds to greedy routing described in Section 2.1.1.1. If no such neighbor exists, i.e., the message is stuck in a local minimum, the protocol switches into the *recovery* mode and tries to escape that local minimum, such that the greedy mode can be used again.

One popular recovery strategy is traversing faces of a plane spanning subgraph of the network. This approach was first proposed by Kranakis et al. [KSU99]. Starting at a source node, it traverses the faces intersected by an imaginary line from source to destination and guarantees delivery for all plane graphs. The first proposed protocols that alternate between greedy and face routing have been Greedy Perimeter Stateless Routing (GPSR) by Karp and Kung [KK00] and Greedy-Face-Greedy (GFG) by Bose et al. [Bos+01]. Assuming the network is modelled by a *unit disk graph*, i.e., every pair of nodes is connected by an edge if and only if their distance is at most 1, a plane spanning subgraph can be computed as a *Gabriel Graph* or a *Relative Neighborhood Graph*, and every node can compute its incident edges in this subgraph locally. Alternatively, local Delaunay triangulation can be used [Gao+05]. Another combination of greedy and face routing is the GOAFR⁺ algorithm by Kuhn et al. [KWZ08b], which has *stretch* $O(c^2)$, i.e., it is guaranteed to reach the destination with cost $O(c^2)$, where c is the minimum cost of a path from source to destination. During the recovery phase, the algorithm restricts face routing to an ellipse and then doubles its radius iteratively. The authors also show that cost $\Omega(c^2)$ is worst-case optimal for any geometric routing algorithm under certain assumptions. A thorough overview of algorithms based on combinations of greedy and face routing as well as implementation details that are essential to guarantee delivery is given by Frey and Stojmenovic [FS06]. Kuhn et al. [KWZ08a] showed that computing a planar subgraph locally works for a superclass of unit disk graphs, namely for certain types of *quasi unit disk graphs*, and proposed geometric routing algorithms with guaranteed delivery for this setting.

In practice, face routing-based approaches may fail to deliver the message due to the fact that for general geometric graphs, the previously mentioned localized strategies for extracting plane spanners might provide graphs that are disconnected or have edge intersections, both of which may cause face routing to fail [Kim+05b]. The Cross-Link Detection Protocol (CLDP) by Kim et al. [Kim+05a] addresses this issue and is able to extract a planar subgraph of any connected network. It sends probe messages that traverse the graph using the right-hand rule and detect link crossings.

An additional problem of face routing is that routes tend to hug the hole boundaries, and due to the resulting uneven load distribution in the network, throughput capacity is reduced [SSG07] and boundary nodes tend to deplete their batteries more quickly than other nodes [GG12].

Leong et al. [LLM06] use a recovery mode that is alternative to face routing. The proposed Greedy Distributed Spanning Tree Routing (GDSTR) algorithm computes and maintains *hull trees*, which are spanning trees of the network in which every tree vertex is annotated by a convex hull of all vertices in its subtree. This information is used when the tree is traversed during the recovery mode. This approach has been extended to the three-dimensional scenario by Zhou et al. [Zho+10].

2.1.1.3 Local routing on geometric graphs

A related line of research considers *local geometric routing* algorithms on geometric graphs [Bos+02; BMo4; DKN10; Bos+15]. In this setting, the nodes are vertices of a geometric graph, and every node u decides which neighbor to forward the message to based on the following information: the destination, a subset of other nodes (typically the neighbors of u), a neighbor v of u that has forwarded the message to u (the *predecessor*) as well as a number of *state* bits stored in the message, which u can modify before sending. A local geometric routing algorithm is *predecessor-oblivious*, if the knowledge of the predecessor is not required, and *predecessor-aware* otherwise. This model generalizes greedy routing, compass routing and greedy-compass routing, all of which are predecessor-oblivious and require no state bits, as well as face routing, which is predecessor-aware and requires $\Theta(\log n)$ state bits for guaranteed delivery on general planar graphs [Bos+15]. For convex subdivisions, face routing only requires predecessor-awareness and the knowledge of the source node to guarantee delivery [KSU99]. Durocher et al. [DKN10] showed that a predecessor-aware local geometric routing algorithm requiring no state bits can not succeed on all geometric *unit ball graphs*, i.e., graphs in which vertices are points in \mathbb{R}^3 and are adjacent if and only if the distance between them is at most 1. For local routing on convex subdivisions with guaranteed delivery, Bose et al. [Bos+15] presented a predecessor-oblivious algorithm requiring one state bit and a predecessor-aware algorithm requiring no state bits. For more results on this topic, we refer to the survey by Durocher et al. [DGW15].

2.1.1.4 Routing with virtual coordinates

An elegant approach proposed to tackle the issues of geographic routing described in Sections 2.1.1.1 and 2.1.1.2 is to assign new, synthetic coordinates to the nodes and then use these *virtual* coordinates for geometric routing [Rao+03; LLM07; Sar+09; Wat+09; Sar+10]. The virtual coordinates are computed using the topology of the network. This is particularly advantageous if no real geographic coordinates are known, for example, if the nodes are not equipped with GPS re-

ceivers. When computing the virtual coordinates, a typical goal is to optimize the success rate of greedy routing. The first such algorithm was NoGeo by Rao et al. [Rao+03]. First, the algorithm identifies perimeter nodes and assigns to them fixed locations in the Euclidean plane that lie on a circle. After that, every non-perimeter node iteratively assigns to itself the center of mass of the current coordinates of its neighbors. This is similar to rubber band embeddings [LLW88] and force-directed graph drawing algorithms [Kob12]. Greedy routing on these virtual coordinates works well in practice, although successful delivery can not be guaranteed. In a similar spirit, Leong et al. [LLMo7] compute virtual coordinates using a system of springs and repulsion forces. In particular, a node s is pushed away from a non-neighbor t , if t is closer to s than any of the neighbors of s . In the approach by Watteyne et al. [Wat+09], the nodes initially have random virtual coordinates which are later updated similarly to the center of mass strategy used in the NoGeo algorithm [Rao+03].

Sarkar et al. [Sar+09] consider dense sensor networks with few holes. Their algorithm extracts a plane mesh from the network and augments it using virtual nodes and edges, such that the union of triangular faces forms a 2-manifold. Then, the authors apply *discrete Ricci flow* [CLo3] to compute a plane straight-line embedding of the resulting mesh, such that every non-triangular face is mapped to a circle. The virtual coordinates are computed using a local gossip-style algorithm. The authors show that a modification of the standard greedy geometric routing guarantees delivery on the resulting embedding, i.e., in some cases, messages might be forwarded to virtual nodes associated with the edges of the mesh. In a later work, Sarkar et al. [Sar+10] achieve improved load balancing by utilizing geometric properties of their network embedding. Intuitively, they bend the routes away from the hole boundaries to improve the battery life of boundary nodes, which otherwise tend to deplete fastest. For further methods that use discrete Ricci flow to compute virtual coordinates for geometric routing, we refer to the survey by Gao et al. [GGL15]. Alternatively, Xia et al. [XWJ14] use discrete *Yamabe flow* and compute embeddings in which hole boundaries are mapped to their convex hulls instead of to circles, which reduces distortion.

Virtual coordinates are not always points in the Euclidean plane. Newsome and Song [NSo3] use polar virtual coordinates. Their approach is based on routing on a spanning tree with additional edges between nodes of the same level. Every subtree of the tree is assigned an angular range that is proportional to the subtree size. These angular ranges are used as virtual coordinates for routing.

Several approaches are based on selecting a subset of nodes as *anchors* or *landmarks* [Car+05; Fan+05; Fon+05; CAo6; LAo6; Ngu+07; FMo7]. Every node in the network computes its hop distances to these landmarks, and the tuples of hop distances are used as virtual coordinates. Greedy routing combined with various recovery strategies is then used on these coordinates. For example, Caruso et al. [Car+05], Cao and Abdelzaher [CAo6] as well as Liu and Abu-Ghazaleh [LAo6] use the Euclidean distance metric for routing, whereas in Beacon Vector Routing by Fonseca et al. [Fon+05], a message is pulled towards landmarks that are closer

to the destination than the current node and pushed away by landmarks that are further away from the destination. Liu and Abu-Ghazaleh [LA06] update the virtual coordinates by averaging among neighbors to increase the success rate of greedy routing, which is similar to the NoGeo algorithm [Rao+03].

2.2 GREEDY EMBEDDINGS

The routing algorithms mentioned in Section 2.1.1.4 aim at computing virtual coordinates on which greedy routing has high success rate. A complementary line of work studies the question, for which network topologies virtual coordinates can be constructed, such that greedy routing has delivery guarantee of 100%. Stated more formally, we want to find out which graphs have a *greedy embedding*. Recall that a greedy embedding of a graph is a mapping of its vertices into a metric space, such that greedy routing on the resulting vertex coordinates using the corresponding distance metric always succeeds; see Chapter 1.

2.2.1 Graphs admitting Euclidean greedy embeddings

The question about the existence of greedy embeddings for various metric spaces and classes of graphs has attracted a lot of interest from the graph drawing and computational geometry communities, the Euclidean plane being the most popular metric space considered. One example of graphs admitting a greedy embedding are Delaunay-realizable graphs, since greedy routing is known to always succeed on Delaunay triangulations [BM04]. Another simple example are graphs with a Hamiltonian path, for example, 4-connected planar graphs [TY94]. Papadimitriou and Ratajczak [PR05] showed that every graph that is planar and 3-connected (i.e., a removal of at most two vertices never disconnects the graph) has a greedy embedding in \mathbb{R}^3 with a custom distance metric that is not the Euclidean distance. They presented a family of graphs that have no greedy embedding in \mathbb{R}^2 with the Euclidean distance metric, namely $K_{k,5k+1}$ (e.g., $K_{1,6}$ is a star with six leaves). Furthermore, they showed that a convex graph drawing in \mathbb{R}^2 in which all angles are at most 120° is greedy. Finally, they conjectured that all 3-connected planar graphs have a greedy embedding in \mathbb{R}^2 with the Euclidean distance metric. Dhandapani [Dha10] proved that every 3-connected planar triangulation has a planar greedy drawing that is a modification of a classical Schnyder drawing [Sch90]. The conjecture by Papadimitriou and Ratajczak itself has been proved independently by Leighton and Moitra [LM10] and Angelini et al. [AFG10]. Both works show this by constructing a greedy drawing for an arbitrary *binary cactus* graph and use the fact that such spanning graph exists for every 3-connected planar graph. Leighton and Moitra [LM10] also gave an example of a binary tree for which no greedy embedding exists. Nöllenburg and Prutkin [NP17] characterized trees admitting a greedy embedding; see Chapter 4 of this thesis. Recently, Da Lozzo et al. [DDF17] showed that every 3-connected planar graph admits a *pla-*

nar greedy embedding. The strong Papadimitriou-Ratajczak conjecture that every 3-connected planar graph admits a *convex* greedy embedding still remains open.

2.2.2 Non-Euclidean greedy embeddings

Kleinberg [Kle07] showed that every connected graph has a greedy embedding in the hyperbolic plane. He also described a distributed algorithm, using which every node can compute its coordinates in such an embedding. The algorithm is based on distributed computation of a rooted spanning tree. Flury et al. [FPW09] construct greedy embeddings of *combinatorial unit disk graphs* (unit disk graphs without geometric information) in spaces of $O(\log^2 n)$ dimensions with bounded hop stretch, i.e., edge counts in paths resulting from greedy routing with the proposed virtual coordinates exceed the lengths of the corresponding shortest paths by at most a constant factor. Ben Chen et al. [Ben+11] present a greedy embedding scheme for 3-connected planar graphs with a non-Euclidean routing metric based on power diagrams. The virtual coordinates are computed in a distributed fashion using the Thurston algorithm for computing a circle packing [Thu85].

2.2.3 Succinctness

Since efficient use of storage and bandwidth are crucial in wireless sensor networks, virtual coordinates should require only few, i.e., $O(\log n)$, bits in order to keep message headers small. Greedy drawings with this property are called *succinct*. The constructions by Kleinberg [Kle07], Leighton and Moitra [LM10] and Angelini et al. [AFG10] do not guarantee succinctness, and the resulting virtual coordinates may require high precision in order to be represented explicitly. For example, when using the hyperbolic embedding by Kleinberg [Kle07], Xia et al. [XW14] observed routing errors in their simulations caused by insufficient precision (using 64-bit doubles and networks with less than 300 000 nodes). Angelini et al. [ADF12] showed that greedy drawings of trees sometimes require exponential area. Eppstein and Goodrich proved the existence of greedy drawings for any connected graph in the hyperbolic plane [EG11], in which virtual coordinates can be encoded succinctly, and Goodrich and Strash [GS09] showed it for 3-connected planar graphs in \mathbb{R}^2 . Wang and He [WH14] used a custom distance metric and constructed convex, planar and succinct drawings for 3-connected planar graphs using Schnyder realizers [Sch90]. In the approach by Flury et al. [FPW09], virtual coordinates require $O(\log^3 n)$ bits. Zhang and Govindaiah [ZG13] construct greedy embeddings into a semi-metric space that consists of tuples of integers between 1 and $2n - 2$. Such virtual coordinates are computed by a simple traversal of a spanning tree, and every tuple consists of at most Δ integers, Δ being the maximum degree of the tree. In this way, it is possible to construct $O(\log n)$ bit virtual coordinates for 3-connected planar graphs or any graphs with a spanning tree that has constant maximum degree.

Succinct greedy embeddings can be considered a special case of *compact routing schemes* [TZo1]. In that setting, every node is labeled using a small, typically polylogarithmic, number of bits. The label of the destination is stored in the message header, possibly along with some additional information, and routing decisions are made locally at every node based on the message header and a precomputed routing table of the node. For example, by storing routing tables of size $\tilde{O}(n^{1/k})$ at every node and using $O(k \log^2 n)$ bit labels, routing with stretch $4k - 5$ can be achieved [TZo1] (\tilde{O} hides a polylogarithmic factor). For an overview of related results, we refer the reader to the survey by Chechik [Che14].

A *labelling scheme for ancestor queries* of a rooted tree is an assignment of labels to the tree nodes, such that the labels of two nodes u and v are sufficient to determine whether u is an ancestor of v in constant time [KNR92]. Such a labeling can be used for local routing on the tree. Dahlgaard et al. [DKR15] presented a labelling scheme for ancestor queries with labels of size $\log_2 n + 2 \log_2 \log_2 n + 3$. Such labels can be viewed as succinct virtual coordinates for local routing.

2.3 GRAPH DRAWINGS WITH GEODESIC-PATH TENDENCY

Finding paths in a graph embedding that always make progress towards their destination, as is the case for greedy routing considered in Section 2.1.1.1, is motivated not only by geographic routing in wireless ad hoc and sensor networks. Studies have shown that such paths are easier to trace for users when exploring a graph drawing. Using eye tracking, Huang et al. [HE05; Hua07] discovered that people exhibit *geodesic-path tendency*, i.e., when eyes encounter nodes with more than one link, the link that goes towards the node is more likely to be searched first. This tendency has been validated by user experiments, in which the task was to find shortest paths in a graph drawing [HEH09]. For example, it has been shown that dead-ends that go towards the target node slow down graph reading. Another notion that has been shown to be important for the readability of graph drawings is *path continuity*, i.e., smooth, continued paths are traced easier than zigzags or paths that detour [War+02; Pur+12]. Not surprisingly, graph drawings in which a path with certain geometric properties exists between every pair of vertices have become a popular research topic. Over the last years a number of different drawing conventions implementing the notion of strong geodesic-path tendency and path continuity have been suggested, namely the aforementioned greedy drawings [Rao+03] as well as (*strongly*) *monotone drawings* [Ang+12], *self-approaching* and *increasing-chord drawings* [Ala+13].

2.3.1 Monotone drawings

While getting closer to the destination in each step, a distance-decreasing path in a greedy drawing can make numerous turns and may even look like a spiral, which hardly matches the intuitive notions of geodesic-path tendency and path continu-

ity. To overcome this, Angelini et al. [Ang+12] introduced monotone drawings, where one requires that for every pair of vertices s and t there exists a *monotone path*, i.e., a path that is monotone with respect to some direction. Ideally, that monotonicity direction should be \vec{st} . This property is called *strong monotonicity*.

Angelini et al. [Ang+12] showed that every tree has a monotone drawing on a grid of area $O(n^{1.6}) \times O(n^{1.6})$ or $O(n) \times O(n^2)$. He and He [HH17] showed that the grid area can be reduced to $12n \times 12n$, which is asymptotically optimal. Oikonomou and Symvonis [OS17] improved the grid area further to $n \times n$. Arkin et al. [ACM89] studied the problem of finding monotone paths between a pair of points among a set of disjoint obstacles and showed that such path always exists if all obstacles are convex. This implies that all *strictly convex* graph drawings (i.e., every face is a convex polygon without flat angles) are monotone [Ang+12; HH15]. Angelini et al. [Ang+12] showed that biconnected planar graphs admit planar monotone drawings, and Hossain and Rahman [HR15] showed that this is the case for all planar graphs. He and He [HH15] showed that the classical Schnyder drawings of 3-connected planar graphs are monotone, even though they are not always strictly convex.

The question of finding plane monotone drawings that preserve the planar embedding of the input graph has also been studied. In this setting, Angelini et al. [Ang+15] proved that all plane graphs admit plane monotone drawings with few bends and that in the special case of biconnected embedded planar graphs and outerplane graphs, there exist plane monotone drawings with straight lines.

In a monotone drawing, the directions with respect to which a monotone path exists might be different for different pairs of vertices. To make the task of finding paths easier for a user, the direction of monotonicity should be easy to determine. Therefore, it is desirable to limit the number of such possible monotonicity directions. He and He [HH15] considered the classical Schnyder drawings of 3-connected planar graphs and showed that there exist six fixed intervals of directions, such that between every pair of vertices there exists a path that is monotone with respect to all directions of one of the intervals. Angelini [Ang17] also considered the classical Schnyder drawings and showed that for 3-connected planar graphs, three monotonicity directions are sufficient, and two are sufficient in Schnyder drawings of maximal planar graphs. In the same work, graphs for which a single monotonicity direction suffices have been characterized.

In strongly monotone drawings, for every pair of vertices s, t there exists a path that is monotone with respect to the direction \vec{st} . Kindermann et al. [Kin+14] showed that every tree admits a strongly monotone drawing and, therefore, so does every connected graph, if crossings are allowed. Felsner et al. [Fel+16] showed that planar strongly monotone drawings exist for 3-connected planar graphs, outerplanar graphs and 2-trees.

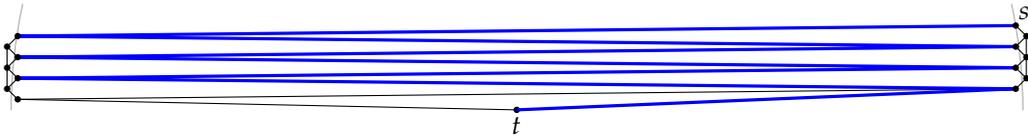


Figure 10: The thick blue zigzag path is the shortest distance-decreasing st -path in the greedy drawing. Figure from a joint work with Angelini et al. [Ang+18].

2.3.2 Geometric spanners

Distance-decreasing paths in a greedy drawing as well as monotone paths may have arbitrarily large detour, i.e., the ratio between the geometric length of a path and the distance of its endpoints can, in general, not be bounded by a constant. Bounding the detour is a popular objective in the area of geometric network design. Given a set of points in the plane, the task is to connect the points with few edges, such that every pair of points in the network is connected by a path with bounded detour. Geometric networks with this property are called *spanners*, and the maximum detour of a shortest path between a pair of vertices in a geometric network is called *dilation*. Chew [Che89] used a variant of Delaunay triangulation and was the first to show that planar spanners with bounded dilation exist for every point set. The standard Euclidean Delaunay triangulation is also a planar spanner [DFS90]. For an overview of the various techniques to construct planar geometric spanners, we refer to the comprehensive surveys by Eppstein [Epp00], Narasimhan and Smid [NS07] and Bose and Smid [BS13]. Spanners in which paths with bounded detour can be found by local routing (see Section 2.1.1.3) have also been studied [Bos+12; Bon+17]. Schindelhauer et al. [SVZ07] considered *weak spanners* and *power spanners*, which are relaxations of geometric spanners. For some constant c and for every pair of vertices s, t , in a weak c -spanner there exists an st -path that remains within a circle around s with radius $c|st|$. For constants c and δ , in a (c, δ) -power spanner, there exists an st -path for every pair of vertices s, t , such that the sum of the δ th powers of the path's edge lengths is at most $c|st|^\delta$. The authors showed that weak spanners are power spanners, but not necessarily vice versa. It is easy to see that greedy drawings are weak 2-spanners and, consequently, power spanners. To the best of my knowledge, it is still open whether greedy drawings are geometric spanners. In Chapter 5, I show that this is the case for greedy drawings of trees and cactuses.

2.3.3 Self-approaching and increasing-chord drawings

Motivated by the notion of bounded detour, Alamdari et al. [Ala+13] initiated the study of *self-approaching* graph drawings. Self-approaching curves, introduced by Icking et al. [IKL99], are curves where for any point t' on the curve, the distance to t' is continuously non-increasing while traversing the curve from the start to t' . Equivalently, a curve is self-approaching if, for any three points a, b, c in this order along the curve, we have $|ac| \geq |bc|$. An even stricter requirement are

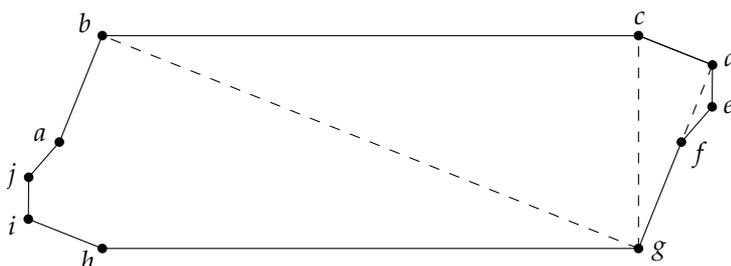


Figure 11: A self-approaching graph drawing that is not monotone. Neither one of the two a - f -paths is monotone in any direction. Dashed lines are edge normals.

so-called *increasing-chord* curves, which are curves that are self-approaching in both directions. The name is motivated by the characterization of such curves, which states that a curve has increasing chords if and only if for any four distinct points a, b, c, d in that order, we have $|bc| \leq |ad|$. Self-approaching curves have detour at most 5.333 [IKL99], and increasing-chord curves have detour at most 2.094 [Rot94]. Note that in greedy drawings, bounding the detour of the shortest distance-decreasing path between a pair of vertices by a constant is impossible in general; see Figure 10.

Alamdari et al. [Ala+13] gave a complete characterization of trees admitting self-approaching drawings. Nöllenburg et al. [NPR16] showed that every triangulation admits a (not necessarily planar) increasing-chord drawing and that every planar 3-tree admits a planar increasing-chord drawing; see Chapter 5. Note that deciding whether two vertices are connected by a self-approaching path in a straight-line graph drawing is NP-hard for three-dimensional drawings [Ala+13] and is conjectured to be NP-hard in two dimensions as well [Bah+17]. Thus, unlike for greedy drawings, recognizing whether a given graph drawing is self-approaching might be NP-hard. Furthermore, Alamdari et al. [Ala+13], Frati et al. [DFG15] and Mastakas and Symvonis [MS15] investigated the problem of connecting given points to obtain an increasing-chord drawing. A special case of increasing-chord graph drawings are *angle-monotone* graph drawings, in which every pair of vertices is joined by a path that is, after some rotation, both x - and y -monotone [Bon+16; LO17].

Every increasing-chord drawing is self-approaching as well as strongly monotone [Ala+13], but a strongly monotone drawing is not necessarily self-approaching. A self-approaching drawing is greedy, but not necessarily monotone (see Figure 11), and a greedy drawing is generally neither self-approaching nor monotone. Greedy drawings of trees are monotone (Angelini et al. [ADF12] showed that such drawings are *slope-disjoint*, which implies monotonicity [Ang+12]). Furthermore, for trees, the notions of self-approaching and increasing-chord drawing coincide since all paths are unique. An overview of existence results for a selection of popular and important classes of planar graphs is given in Table 1. To the best of my knowledge, no graphs are known that admit a self-approaching drawing, but no increasing-chord drawing.

Table 1: An overview of existence results of greedy, monotone, strongly monotone and increasing-chord drawings for popular classes of planar graphs. The highlighted results are presented in Chapters 4 and 5 of this thesis.

	greedy	monotone	strongly monotone	increasing-chord
trees	characterized [NP17]	always [Ang+12]	always [Kin+14]	characterized [Ala+13]
binary cactuses	always, planar [LM10; AFG10]	always, even all outerplane, preserve embedding [Ang+15]	always, even all outerplane, convex and preserve embedding [Fel+16]	not always [NPR16]
triangulations	always, planar [Dha10]	always, planar [ACM89; HH15; Ang17]	always, planar [Fel+16]	always [NPR16]; planarity in special cases, open in general
3-connected planar graphs	always [LM10; AFG10], planar [DDF17], convexity open	always, planar [HH15; Ang17]	always, planar [Fel+16]	open
connected planar	not always [PRO5]	always, planar [HR15]	planarity: not always, open for biconnected [Fel+16]	not always [PRO5]

2.4 GREEDY GEOMETRIC ROUTING IN CONTINUOUS DOMAINS

Let us now return to wireless sensor networks and consider a network of sensors distributed over a closed region. An assumption often made in the literature is that the distribution of the sensors is very dense, except for relatively few holes [Sar+09; TK12; Bir13]. Assuming the density of nodes within the network boundary is close to infinity, greedy routing will forward a message along the straight line towards its destination or, if the boundary is hit, the message will slide along the boundary, as long as it decreases the Euclidean distance to the destination.

2.4.1 Beacon-based routing

For the above setting, Biro et al. [Bir+11] proposed the *beacon-based routing* model. A message is modeled by a point that moves inside a polygonal region \mathcal{P} , inside which there exists a set of beacons. When activated, a beacon creates a magnetic pull everywhere inside \mathcal{P} , such that a point in \mathcal{P} either moves towards the beacon along the straight line, or, if the boundary is hit, slides along the boundary, as long as the distance to the beacon decreases continuously. Once decreasing the distance to the beacon is no longer possible, the point gets stuck. Only one beacon is active at each point in time, and when it is reached by the moving point, the beacon is deactivated, and another one can be activated. Points s, t are *routed* if there exists a sequence of beacons ending with t , such that the beacons are activated consecutively and such that every currently active beacon is reached by s eventually and then deactivated. The authors studied the complexity of covering, or guarding, polygonal regions with few beacons, such that all pairs of points are routed. Here, the beacons for the destinations are not counted, since otherwise every point in the region must be a beacon. In a follow-up work, Biro et al. [Bir+13] designed algorithms to select a minimum sequence of beacons to forward a message to a given destination point. Various routing and guarding problems in the beacon-based routing model were covered in detail by Michael Biro in his dissertation [Bir13].

Beacon-based routing is related to landmark-based techniques for routing in wireless sensor networks mentioned in Section 2.1.1.4. Recall that in these techniques, routing decisions at every node are made based on distances to a subset of designated landmark nodes. For example, in Gradient Landmark-Based Distributed Routing (GLIDER) by Fang et al. [Fan+05], the network is partitioned into Voronoi cells of the landmark nodes, i.e., all nodes inside a cell have a closest landmark in common with respect to hop distance. The adjacency graph of the cells is used to route messages to a different cell, whereas greedy routing on virtual coordinates is used for intra-cell routing.

2.4.2 Network decomposition for routing

Similar to GLIDER [Fan+05], other approaches decompose the network into components such that in each of them greedy routing or variants thereof are likely to perform well [BGJ07; FM07; TBK09; ZSG09; TK12; Jia+15]. A global data structure of preferably small size is used to store interconnectivity between components. One such network decomposition approach proposed by Tan and Kermarrec [TK12] will be considered in detail here. The authors assume that *global* connectivity irregularities, i.e, large holes in the network and the network boundary, are the main source of local minima in which greedy routing between a pair of sensor nodes might get stuck. They note that in practical sensor networks, *local* connectivity irregularities normally have low impact on the cost of routing and the quality of the resulting paths, since the local minima in this context can be overcome by simple and light-weight techniques; see [TK12] for a list of such strategies. With this reasoning, Tan and Kermarrec model the network as a polygonal region with obstacles or holes inside it and consider greedy routing inside this continuous domain, similarly to the beacon-based routing model proposed by Biro et al. [Bir+11]. Local minima now only appear on the boundaries of the polygonal region. In Chapter 6, the same model is used.

Tan and Kermarrec [TK12] try to partition this region into a minimum number of polygons, in which greedy routing works between every pair of points. They call such components *greedily routable regions* (GRRs). For intercomponent routing, region adjacencies are stored in a graph. In the continuous setting, the algorithm is able to guarantee finding paths with bounded detour.

For routing in the underlying network of sensor nodes corresponding to discrete points inside the polygonal region, greedy routing is used if the source and the destination nodes are in the same component, and existing techniques are used to overcome local minima. For inter-component routing, every node stores a neighbor on a shortest path to each component. This information is used to get to the component of the destination, and then intra-component routing is used.

Tan and Kermarrec [TK12] emphasize the importance for the nodes to store as small routing tables as possible and note that the size of a node's routing table directly reflects the number of network components in a decomposition. Therefore, the goal is to partition the network into a minimum number of GRRs. The authors prove that partitioning a polygonal region with holes into a minimum number of GRRs is NP-hard and propose a simple heuristic. Its solution may strongly deviate from the optimum even for very simple polygons; see the examples in Chapter 6.

The problem of partitioning a polygonal region into a minimum number of GRRs is strongly reminiscent of partitioning a polygonal region into a minimum number of convex subpolygons, which is a well-studied problem from computational geometry. For an overview of the results on the convex partition problem, see the survey by Keil [Kei00]. For polygonal regions with holes, minimum convex partition is known to be NP-hard if *Steiner points* are allowed (i.e., cuts of the



Figure 12: A benchmark instance of the GRR decomposition problem. Figure taken from the work of Tan and Kermarrec [TK12]. (a) A network of streets over which wireless sensor nodes are densely distributed. (b) The resulting network is approximated as a thin polygonal region and partitioned into GRRs.

partition are not necessarily diagonals of the input polygonal region) [Lin82], as well as if no Steiner points are allowed [Kei85]. Therefore, it is not completely unexpected that the minimum GRR partition problem for polygonal regions with holes is NP-hard as well. For simple polygons without holes, however, minimum convex partitions both with and without Steiner points can be computed in polynomial time [Kei85; CD85], and approximate solutions can be computed using simple strategies. For example, the naive strategy of iteratively cutting along the bisectors of reflex angles provides an approximation of factor 2 for the minimum partition with Steiner points of simple hole-free polygons [CD85]. It is therefore surprising that for minimum GRR partition of simple polygons without holes, no polynomial-time optimal solutions or even constant-factor approximations were proposed.

Some benchmark instances from the work of Tan and Kermarrec [TK12, Figure 17] are networks of sensor nodes distributed on roads of a city; see Figure 12. The resulting polygonal regions are very narrow and strongly resemble plane straight-line graph drawings. Therefore, considering plane straight-line graph drawings in addition to polygonal regions is a natural adjustment of the minimum GRR partition problem. In this scenario, GRRs coincide with increasing-chord drawings of trees as studied by Alamdari et al. [Ala+13]. I approach this problem in Chapter 6 of this thesis.

Recently, Bose et al. [BKL17] studied the complexity of finding shortest self-approaching paths in simple polygons. Additionally, the authors characterized self-approaching polygons, i.e., polygons in which every pair of points can be connected by a self-approaching path, and provided a linear-time algorithm to recognize such polygons. From this characterization, it follows that self-approaching

polygons are exactly the greedily routable regions studied by Tan and Kermarrec [TK12] and in Chapter 6 of this thesis.

3

PRELIMINARIES

This chapter recalls several graph-theoretic concepts and defines notation that will be used in this thesis.

3.1 GRAPHS, PATHS AND CONNECTIVITY

A *graph* is a mathematical object that is often used to model relationships between entities. Formally, a graph is a tuple $G = (V, E)$, where $V = \{v_1, \dots, v_n\}$ is a set of *vertices* and $E = \{e_1, \dots, e_m\}$ a set of edges, such that every edge $e \in E$ is a tuple $e = \{u, v\}$ for vertices $u, v \in V$. For brevity, we write uv instead of $\{u, v\}$. We call a pair of vertices $u, v \in V$ *adjacent*, if $uv \in E$. For an edge $e = uv \in E$, we say that u and v are *endpoints* of e and that both u and v are *incident* to e . Edges e_1, e_2 are incident to each other, if they share a common endpoint. The *degree* $\deg(v)$ of a node v is the number of edges incident to v in G . We call vertices adjacent to a vertex $v \in V$ the *neighbors* of v . Let $N(v)$ denote the neighbors of v in G .

graph

A *path* in G is a tuple $\rho = (v_0, v_1, \dots, v_k)$, such that $v_i v_{i+1} \in E$ for $i = 0, \dots, k-1$. We say that ρ has *length* k . For vertices $s, t \in V$, such that $s = v_0$ and $t = v_k$, we call such a tuple (v_0, v_1, \dots, v_k) an *s-t-path* (or *st-path*) in G . A path is *simple*, if v_i are pairwise distinct, for $i = 0, \dots, k$.

s-t-path

We say that $s, t \in V$ have *distance* k in G , if the minimum length of an *st-path* in G is k . We write $d_G(s, t) = k$. For $k \in \mathbb{N}$, $k \geq 1$, the *k-neighborhood* $N_k(v)$ of a node $v \in V$ is defined as $N_k(v) = \{u \in V \mid d_G(v, u) \leq k\}$.

distance $d_G(s, t)$
in G

For a path $\rho = (v_0, v_1, \dots, v_k)$ and $0 \leq p \leq q \leq k$, we call (v_p, \dots, v_q) a *subpath* of ρ . A path (v_0, v_1, \dots, v_k) , $k \geq 3$, is a *cycle*, if $v_0 = v_k$. A cycle (v_0, v_1, \dots, v_k) is *simple*, if for $i = 1, \dots, k-1$, all vertices v_i are pairwise distinct.

A graph $G_1 = (V_1, E_1)$ is a *subgraph* of a graph $G = (V, E)$, if we have $V_1 \subseteq V$ and $E_1 \subseteq E$. Such a subgraph G_1 is a *spanning* subgraph of G , if $V_1 = V$.

For a pair of vertices $s, t \in V$, we say that s and t are *connected in* G if an *st-path* exists in G . A graph G is *connected*, if every pair of its vertices is connected in G . A *connected component* is an inclusion-maximal connected subgraph of G .

If no cycle exists in G , then G is a *forest*. A *tree* is a connected forest. A *subtree* is a subgraph of a tree that is a tree itself. A *leaf* of a tree is a tree vertex of degree 1.

For a graph $G = (V, E)$ and vertex set $V' \subseteq V$, let $G - V'$ denote the graph with vertex set $V \setminus V'$ and edge set $\{uv \in E \mid u, v \in V \setminus V'\}$. For an edge set $E_1 \subseteq E$,

$G - V', G \pm E'$

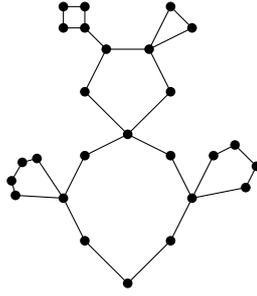


Figure 13: A binary or Christmas cactus graph.

let $G - E_1$ denote the graph with vertex set V and edge set $E \setminus E_1$. For an edge set E_2 , let $G + E_2$ denote the graph with vertex set

$$V \cup \{u \mid u \text{ is an endpoint of an edge in } E_2\}$$

and edge set $E \cup E_2$. We write $G - v$ instead of $G - \{v\}$ and $G \pm uv$ instead of $G \pm \{uv\}$.

Let $G = (V, E)$ be a connected graph. A *separating k -set* is a set $S \subseteq V$ of k vertices whose removal disconnects the graph, i.e., $G - S$ is disconnected. A vertex forming a separating 1-set is called *cutvertex*. A graph is *c -connected* if it does not admit a separating k -set with $k \leq c - 1$; 2-connected graphs are also called *biconnected*. A connected graph is biconnected if and only if it does not contain a cutvertex. A *block* is a maximal biconnected subgraph.

A *subdivision* of a graph $G = (V, E)$ is a graph created by replacing every edge in a set $E_1 \subseteq E$ by a simple path.

A *caterpillar* is a tree T , such that the graph created by removing all leaves of T is a path. A tree is a *star* if it has exactly one vertex v with $\deg(v) \geq 2$. A star with d leaves is denoted by $K_{1,d}$.

A *rooted tree* T is a tree $T = (V, E)$ together with a distinguished vertex $r \in V$ called *root*. A *subtree* of a node v is then the subtree of T with root v containing all vertices u , for which the u - r -path in T contains v . A *parent* of $v \in V$ is a neighbor u of v that is not in the subtree of v .

A *cactus* is a graph in which every edge is part of at most one simple cycle. A *binary* (or *Christmas*) cactus is a cactus in which every vertex is part of at most two blocks; see Figure 13.

3.2 GRAPH DRAWINGS AND POLYGONS

A *drawing* Γ of a graph $G = (V, E)$ maps every vertex $v \in V$ to a point $\Gamma(v) \in \mathbb{R}^2$ and every edge $uv \in E$ to a simple open curve $\Gamma(uv) \subseteq \mathbb{R}^2$ with endpoints $\Gamma(u)$ and $\Gamma(v)$.

A graph drawing is *planar* if the curves intersect only at their common endpoints, i.e., 1) curves $\Gamma(uv), \Gamma(xy)$ have an intersection only if edges $uv, xy \in E$ are incident in G and 2) for edges uv and uw , the only intersection of $\Gamma(uv)$

and $\Gamma(uw)$ is $\Gamma(u)$. A graph is *planar* if it admits a planar drawing. A graph is *maximal planar* or a *triangulation* if no edge can be added to it without violating the graph's planarity.

triangulation

A drawing is *straight-line* if every $\Gamma(uv)$ is a straight-line segment. In this thesis, we only consider graph drawings that are straight-line. If the drawing Γ is fixed, we associate every vertex $v \in V$ with the corresponding point $\Gamma(v)$ and every edge uv with the corresponding straight-line segment $\Gamma(uv)$. Straight-line graph drawings are also called *geometric graphs*.

A *simple polygon* (or just *polygon*) is a closed flat region bounded by a closed chain of straight-line segments that has no self-intersections. A *polygonal region* is defined by a simple polygon and $m \geq 0$ holes contained inside it, each of which is a simple polygon.

simple polygon
polygonal region

3.2.1 Greedy, monotone, self-approaching and increasing-chord drawings

For points $p, q \in \mathbb{R}^2$, let $|pq|$ denote the Euclidean distance between p and q .

distance $|pq|$

Definition 3.1 (Distance-decreasing path). Consider a straight-line graph drawing Γ and vertices s, t in Γ . An st -path $(s = v_0, v_1, \dots, v_k = t)$ is *distance-decreasing*, if we have $|v_{i+1}t| < |v_it|$ for every $i = 0, \dots, k - 1$.

distance-decreasing path

Definition 3.2 (Greedy embedding). A straight-line graph drawing Γ of a graph $G = (V, E)$ is a *greedy embedding* or a *greedy drawing*, if for every pair of vertices $s, t \in V, s \neq t$, there exists a distance-decreasing st -path in Γ .

greedy embedding

Equivalently, a straight-line drawing Γ of a graph $G = (V, E)$ is a greedy drawing if for every pair of vertices $s, t \in V, s \neq t$, there exists a neighbor u of s with $|ut| < |st|$.

Definition 3.3 (Self-approaching curve [IKL99]). An oriented curve is *self-approaching* if, for any three points a, b, c in this order along the curve, we have $|ac| \geq |bc|$.

Definition 3.4 (Increasing-chord curve [Rot94]). An oriented curve is *increasing-chord* (or has increasing chords) if, for any four points a, b, c, d in this order along the curve, we have $|bc| \leq |ad|$.

An st -path in a straight-line graph drawing Γ is *self-approaching* (increasing-chord), if the corresponding curve is self-approaching (increasing-chord).

self-approaching and increasing-chord paths

Definition 3.5 (Self-approaching and increasing-chord graph drawings [Ala+13]). A straight-line drawing Γ of a graph $G = (V, E)$ is *self-approaching* (increasing-chord), if every vertex pair $s, t \in V$ is joined by a self-approaching (increasing-chord) st -path.

Definition 3.6 (Monotone and strongly monotone graph drawings [Ang+12]). A straight-line drawing Γ of a graph $G = (V, E)$ is

1) *monotone*, if every vertex pair $s, t \in V$ is joined by an st -path that is monotone in some direction;

2) strongly monotone, if every vertex pair $s, t \in V$ is joined by an st -path that is monotone in direction \vec{st} .

Definition 3.7 (Detour and dilation). 1. For an oriented curve C and points p, q on C , the detour of C between p and q is defined as $\frac{d_C(p,q)}{|pq|}$, where $d_C(p,q)$ denotes the distance from p to q along the curve C .

2. For vertices s, t of a geometric graph Γ , the detour of Γ between s and t is the minimum detour between s and t of a curve corresponding to an st -path in Γ .

3. The vertex dilation of a geometric graph Γ is the largest detour from s to t on Γ over all pairs of vertices s, t of Γ .

In this thesis, *dilation* of a geometric graph Γ stands for vertex dilation. Other terms used for dilation in the literature are *spanning ratio* or *stretch factor*. A geometric graph whose dilation is bounded by a constant c is a *geometric c -spanner*.

3.2.2 Used geometric notation

Let \vec{e}_1 and \vec{e}_2 be the first and the second standard basis vectors. We say that vector \vec{d} points upwards if the dot product $\vec{d} \cdot \vec{e}_2$ of \vec{d} and \vec{e}_2 is positive. We say that \vec{d} points downwards if we have $\vec{d} \cdot \vec{e}_2 < 0$. We say that \vec{d} points to the right if we have $\vec{d} \cdot \vec{e}_1 > 0$. We say that \vec{d} points to the left if we have $\vec{d} \cdot \vec{e}_1 < 0$.

For the ease of notation, we shall sometimes treat a graph edge uv as an ordered tuple. For example, for a given straight-line graph drawing Γ , we say that an edge uv points upwards if the vector \vec{uv} points upwards, etc.

For points $a, b, c \in \mathbb{R}^2$, let $\angle abc$ denote the angle at b formed by ab and bc , such that $\angle abc \leq 180^\circ$. For vectors \vec{ab}, \vec{cd} , let $\angle_{\text{ccw}}(\vec{ab}, \vec{cd})$ denote the counterclockwise turn (or turning angle) from \vec{ab} to \vec{cd} . Analogously, let $\angle_{\text{cw}}(\vec{ab}, \vec{cd})$ denote the clockwise turn from \vec{ab} to \vec{cd} . We say that point p is inside the angle $\angle abc < 180^\circ$, if either $\angle_{\text{cw}}(\vec{ba}, \vec{bp}) \leq \angle_{\text{cw}}(\vec{ba}, \vec{bc}) < 180^\circ$ or $\angle_{\text{ccw}}(\vec{ba}, \vec{bp}) \leq \angle_{\text{ccw}}(\vec{ba}, \vec{bc}) < 180^\circ$. Otherwise, we say that p is outside of $\angle abc$.

For points $a, b \in \mathbb{R}^2$, let $\text{ray}(a, b)$ denote the ray with origin a and direction \vec{ab} . For a vector \vec{d} , let $\text{ray}(a, \vec{d})$ denote the ray with origin a and direction \vec{d} .

Part I

GRAPH EMBEDDINGS MOTIVATED BY GREEDY
ROUTING: EXISTENCE

4

EUCLIDEAN GREEDY DRAWINGS OF TREES

In this chapter, we¹ completely characterize the trees that admit a greedy embedding in \mathbb{R}^2 . This answers a question by Angelini et al. [ADF12] and is a further step in characterizing the graphs that admit Euclidean greedy embeddings. Chapter 4 is based on joint work with Martin Nöllenburg [NP13; NP17].

4.1 INTRODUCTION

The question about the existence of greedy embeddings for various metric spaces and classes of graphs has attracted a lot of interest. An overview of related work is presented in Chapter 2 of this thesis. We recall that most works on Euclidean greedy embeddings consider 3-connected planar graphs [PR05; AFG10; LM10; Dha10; GS09; DDF17].

4.1.1 Contribution

We give the first complete characterization of all trees that admit a greedy embedding in \mathbb{R}^2 with the Euclidean distance metric. This solves the corresponding open problem stated by Angelini et al. [ADF12] and is a further step in characterizing the graphs that have greedy embeddings. For any given tree T and an edge e of T separating T into T_1 and T_2 , we calculate a tight upper bound on the opening angle of a cone formed by perpendicular bisectors of edges of T_1 , in which T_2 is contained in any greedy embedding in time linear in the size of T_1 . We then show that deciding whether T has a greedy embedding is equivalent to deciding whether there exists a valid angle assignment in a certain wheel polygon. This includes a non-linear constraint known as the *wheel condition* [DV96]. For most cases (all trees with maximum degree 4 and most trees with maximum degree 5) we are able to give an explicit solution to this problem, which provides a linear-time recognition algorithm. For trees with maximum degree 3 we give an alternative characterization by forbidden subtrees. For some trees with one degree-5 node we resort to using non-linear solvers. For trees with nodes of degree ≥ 6 no greedy drawings exist.

¹ In the remainder of my thesis, I write “we” instead of “I” for convenience.

Our proofs are constructive, however, we ignore the possibly exponential area requirements for our constructions. This is justified, since some trees require exponential-size greedy drawings as shown by Angelini et al. [ADF12].

4.2 PRELIMINARIES

In this section, we introduce the concept of the opening angle of a rooted subtree and present relations between opening angles that will be crucial for the characterization of greedy-drawable trees. We start with a number of lemmas on basic properties of opening angles and greedy drawings and sketch the main ideas of our characterization. This is followed by proving the *shrinking lemma*, which serves as a main tool for our later constructions.

It is known that for a greedy drawing Γ of T any subtree of T is represented in Γ by a greedy subdrawing [ADF12]. We define the *axis* of an edge uv as its perpendicular bisector. The following property of greedy drawings is easy to show.

Observation 4.1. *In a greedy drawing of tree $T = (V, E)$, for every edge $uv \in E$, $\text{axis}(uv)$ contains no vertices of T .*

Proof. Let vertex w lie on $\text{axis}(uv)$. Either the u - w -path or the v - w -path contains edge uv . In both cases, the distance to w must decrease for the nodes of the path, a contradiction to $|uw| = |vw|$. \square

Let h_{uv}^u denote the open half-plane bounded by the axis of uv and containing u . Let T_{uv}^u be the subtree of T containing u obtained from T by removing uv . The following property is the immediate consequence of Observation 4.1 and Lemma 3 in [ADF12].

Lemma 4.1. *In a greedy drawing of tree $T = (V, E)$, for every edge $uv \in E$,*

- (a) $\text{axis}(uv)$ has empty intersection with every edge from $E \setminus \{uv\}$, and
- (b) every subtree T_{uv}^u is contained in h_{uv}^u .

The converse is also true.

Lemma 4.2. *Let Γ be a drawing of T with $T_{uv}^u \subseteq h_{uv}^u$ for all $uv \in E$. Then, Γ is greedy.*

Proof. For $s, t \in V$ let u be the neighbor of s on the unique s - t -path in T . Since $t \in T_{su}^u \subseteq h_{su}^u$, we have $|ut| < |st|$. \square

Angelini et al. [ADF12] further showed that greedy tree drawings are always planar and that in any greedy drawing of T the angle between two adjacent edges must be strictly greater than 60° . Thus T cannot have a node of degree ≥ 6 .

Lemma 4.3 (Lemma 7 in [ADF12]). *Consider two edges uv and wz in a greedy drawing of T , such that the path from u to w does not contain v and z . Then, the rays $\text{ray}(u, \vec{uv})$ and $\text{ray}(w, \vec{wz})$ diverge; see Figure 14.*

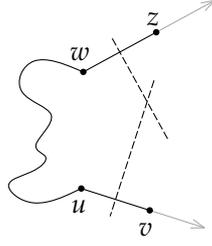


Figure 14: Sketch of Lemma 4.3.

Lemma 4.4. *Let Γ be a greedy drawing of $T = (V, E)$, $v \in V$, $\deg(v) = 2$, $N(v) = \{u, w\}$ the only two neighbors of v , and $T' = (T - v) + uw$. The drawing Γ' induced by replacing segments uv and vw by uw in Γ is a greedy drawing of T' .*

shortcuts

Proof. For x, y in T' , let ρ' and ρ be the x - y -paths in T' and in T , respectively. If $\rho \neq \rho'$, then $v \in \rho$. Since distance to y decreases along ρ , it also decreases along ρ' . Hence, Γ' is greedy. \square

Let Γ be a greedy drawing of T . We shall consider subtrees $T_i = (V_i, E_i)$ of T , such that T_i has root r_i , $\deg(r_i) = 1$ in T_i and v_i is the neighbor of r_i in T_i . Note that T_i is contained in $T_{r_i v_i}^{v_i} + r_i v_i$, but we might have $T_i \subsetneq T_{r_i v_i}^{v_i} + r_i v_i$.

Definition 4.1 (Polygon of a rooted subtree). *We define the polygon of a rooted subtree T_i as $\text{polygon}(T_i) = \bigcap \{h_{uw}^w \mid uw \in E_i, d_T(w, r_i) < d_T(u, r_i)\}$.*

$\text{polygon}(T_i)$

Next, we consider the directions of the edges of T_i in the underlying straight-line drawing. When considering T_i , we assume that edge $v_i r_i$ is oriented from v_i to r_i , and every other edge xy of T_i , $d_T(x, r_i) < d_T(y, r_i)$, is oriented from x to y . Figure 15a shows subtree T_i , and the corresponding edge directions are shown in Figure 15b. The successor and predecessor of $v_i r_i$ in the resulting clockwise order shall play an important role in our constructions.

Definition 4.2 (Extremal edges of a rooted subtree). 1. *Let $|E_i| \geq 2$. The clockwise extremal edge $\overrightarrow{a_1 b_1} \neq \overrightarrow{v_i r_i}$ of T_i , $d_T(a_1, r_i) < d_T(b_1, r_i)$, is the edge for which the direction $\overrightarrow{a_1 b_1}$ is closest to $\overrightarrow{v_i r_i}$ clockwise. Analogously, the counterclockwise extremal edge $\overrightarrow{a_2 b_2} \neq \overrightarrow{v_i r_i}$ of T_i , $d_T(a_2, r_i) < d_T(b_2, r_i)$, is the edge for which the direction $\overrightarrow{a_2 b_2}$ is closest to $\overrightarrow{v_i r_i}$ counterclockwise. We break ties arbitrarily.*

2. *For $|E_i| = 1$, edge $v_i r_i$ is both the clockwise extremal edge and the counterclockwise extremal edge.*

Note that by Lemma 4.3, if $|E_i| \geq 2$, $\text{ray}(a_j, \overrightarrow{a_j b_j})$ and $\text{ray}(v_i, \overrightarrow{v_i r_i})$ diverge for $j = 1, 2$. Moreover, if $\angle_{\text{cw}}(\overrightarrow{a_1 b_1}, \overrightarrow{a_2 b_2}) \leq 180^\circ$, then $\text{polygon}(T_i)$ is obviously unbounded, and its boundary rays are formed by parts of $\text{axis}(a_1 b_1)$ and $\text{axis}(a_2 b_2)$. For example, in Figure 15a, $\text{polygon}(T_i)$ is formed by the axes of $a_1 b_1$, $a_2 b_2$ and $v_i r_i$.

In the following, let $a_1 b_1$ be the clockwise extremal edge of T_i , let $a_2 b_2$ be the counterclockwise extremal edge of T_i , and let $d_T(a_j, r_i) < d_T(b_j, r_i)$ for $j = 1, 2$.

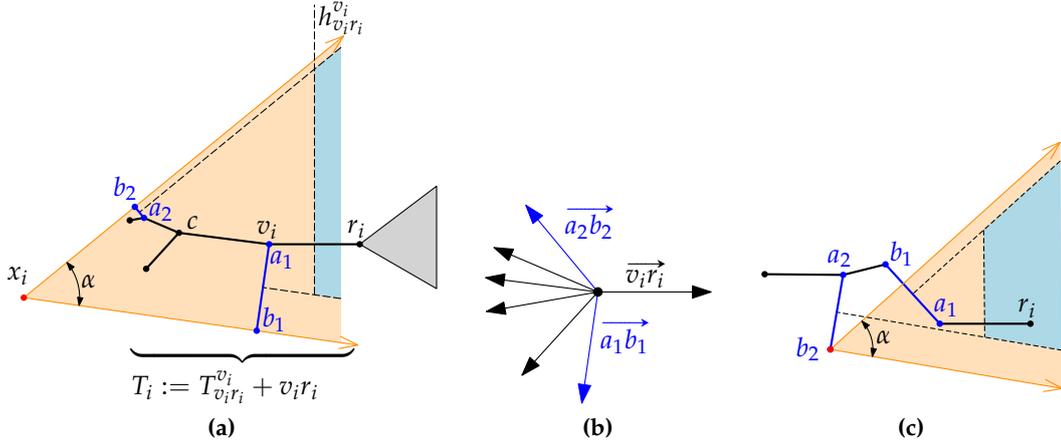


Figure 15: (a) Subtree T_i with opening angle $\angle T_i$ (orange cone), extremal edges $a_1 b_1, a_2 b_2$ (blue), $x_i = \text{apex}(T_i)$ (red) and $\text{polygon}(T_i)$ (light blue). The subtree $T_{v_i r_i}^{v_i}$ (gray triangle) must be contained in the half-plane $h_{v_i r_i}^r$ and the cone $\angle T_i$. (b) Directions of the edges of T_i . (c) Opening angle of T_i and $\text{apex}(T_i)$ for the second case of Definition 4.4.

Definition 4.3 (Open angle). Let $\angle_{\text{cw}}(\overrightarrow{a_1 b_1}, \overrightarrow{a_2 b_2}) < 180^\circ$. Then, $\text{polygon}(T_i)$ is unbounded, and we say that T_i is drawn with an open angle.

Definition 4.4 (Apex).

- (a) If $a_1 b_1$ and $a_2 b_2$ are not on the same path from r_i to a leaf of T_i , we define $\text{apex}(T_i)$ as the intersection of the line through b_1 parallel to axis($a_1 b_1$) and the line through b_2 parallel to axis($a_2 b_2$); see Figure 15a.
- (b) Otherwise, we define $\text{apex}(T_i)$ as the vertex b_j , $j = 1, 2$, such that $d_T(b_j, r_i)$ is maximal; see Figure 15c.

Definition 4.5 (Opening angle). Let T_i be drawn with an open angle. For $j = 1, 2$, let h_j be the half-plane bounded by the line through b_j parallel to axis($a_1 b_1$), such that $a_j \in h_j$. We define $\angle T_i = h_1 \cap h_2$ and call $\angle T_i$ the opening angle of T_i in Γ (orange cone in Figures 15a and 15c.). We write $|\angle T_i| = \alpha$, for $\alpha = 180^\circ - \angle_{\text{cw}}(\overrightarrow{a_1 b_1}, \overrightarrow{a_2 b_2})$.

If $0^\circ < \angle_{\text{cw}}(\overrightarrow{a_1 b_1}, \overrightarrow{a_2 b_2}) < 180^\circ$, then $\angle T_i$ is a cone with apex at $\text{apex}(T_i)$. For uniformity, we say that $\angle T_i$ has apex at $\text{apex}(T_i)$ for $\angle_{\text{cw}}(\overrightarrow{a_1 b_1}, \overrightarrow{a_2 b_2}) = 0^\circ$ as well, even though $\angle T_i$ is a half-plane in that case.

Note that $\text{polygon}(T_i) \subseteq \angle T_i$ by the following observation.

Observation 4.2. Let h be an open half-plane and $p \notin h$. Let h' be the half-plane created by a parallel translation of the boundary of h' to p . Then, $h \subseteq h'$.

Definition 4.6 (Closed and zero angle). Let $\angle_{\text{cw}}(\overrightarrow{a_1 b_1}, \overrightarrow{a_2 b_2}) > 180^\circ$ (or $= 180^\circ$). Let $C_i = h_{a_1 b_1}^{a_1} \cap h_{a_2 b_2}^{a_2}$. Let p_1, p_2 be the midpoints of $a_1 b_1$ and $a_2 b_2$, respectively. We denote the part of C_i bounded by segment $p_1 p_2$ containing r by ∇T_i and say that T_i is drawn with a closed (or zero) angle; see Figure 16a. We write $|\angle T_i| < 0$ (or $= 0$).

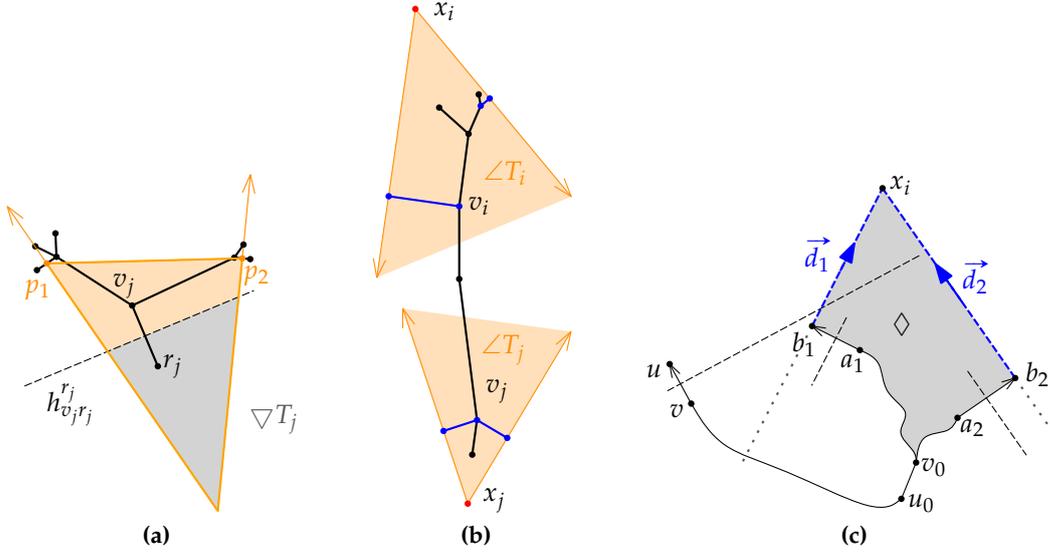


Figure 16: Proof of Lemma 4.5. (a) Subtree T_j with closed angle $\angle T_j$ and boundary segment $p_1 p_2$. (b) Open angles of independent subtrees must contain apices of each other. (c) Proving $\text{apex}(T_i) \in \text{polygon}(T_j)$ using Lemma 4.3.

Note that if $|\angle T_i| \leq 0$, the extremal edges $a_1 b_1$ and $a_2 b_2$ cannot lie on the same path from r_i to a leaf. Otherwise, that path would have a U-turn and, therefore, would not be drawn greedy.

Definition 4.7 (Independent rooted subtrees). *We say that two subtrees T_1, T_2 with roots r_1, r_2 of degree 1 in the corresponding subtree are independent, if $T_2 - r_2 \subseteq T_{v_1 r_1}^{r_1}$ and $T_1 - r_1 \subseteq T_{v_2 r_2}^{r_2}$.*

If T_1 and T_2 are independent, then $T_2 - r_2 \subseteq h_{v_1 r_1}^{r_1}$ and $T_1 - r_1 \subseteq h_{v_2 r_2}^{r_2}$ in Γ by Lemma 4.1. Also, if $r_2 \notin T_{v_1 r_1}^{r_1}$, then $r_2 = v_1$. Figure 16b shows two independent subtrees.

The next lemma describes possible arrangements of the opening angles of independent subtrees.

Lemma 4.5. *Let T_i and T_j be independent subtrees of tree T . Consider a greedy drawing Γ of T .*

- (a) *Let $|\angle T_i|, |\angle T_j| > 0$ in Γ . Then, $\text{apex}(T_i) \in \angle T_j$ and $\text{apex}(T_j) \in \angle T_i$.*
- (b) *Let $|\angle T_i| > 0, |\angle T_j| \leq 0, p_1 p_2$ the boundary segment of $\angle T_j$. Then, $\text{apex}(T_i) \in \nabla T_j$, and $p_1, p_2 \in \angle T_i$.*
- (c) *Let $|\angle T_i|, |\angle T_j| \leq 0, p_1 p_2$ the boundary segment of ∇T_j . Then, $p_1, p_2 \in \nabla T_i$.*

Proof. (a) We prove $\text{apex}(T_i) \in \angle T_j$. Let $a_1 b_1$ and $a_2 b_2$ be the extremal edges of T_i .

If $a_1 b_1$ and $a_2 b_2$ are on the same path from r_i to a leaf of T_i , then, by Definition 4.4, $\text{apex}(T_i)$ is a node of T_i . Then, $\text{apex}(T_i) \in T_i \subseteq \text{polygon}(T_j) \subseteq \angle T_j$.

If a_1b_1 and a_2b_2 are not on the same path from r_i to a leaf of T_i , we have $a_2b_2 \subseteq h_{a_1b_1}^{a_1}$ and $a_1b_1 \subseteq h_{a_2b_2}^{a_2}$. Without loss of generality, we can assume that $\overrightarrow{a_1b_1}$ and $\overrightarrow{a_2b_2}$ point upwards and $\overrightarrow{v_i r_i}$ points downwards. Then, all edges of $T_i - v_i r_i$ point upwards. Let \vec{d}_1 be a vector with $\angle_{\text{cw}}(\overrightarrow{a_1b_1}, \vec{d}_1) = 90^\circ$, and let \vec{d}_2 be a vector with $\angle_{\text{ccw}}(\overrightarrow{a_2b_2}, \vec{d}_2) = 90^\circ$; see the blue arrows in Figure 16c. Then, point $x_i = \text{apex}(T_i)$ is the intersection of ray (b_1, \vec{d}_1) and ray (b_2, \vec{d}_2) .

Let ρ_k be the r_i - b_k -path in T_i and u_0v_0 the last common edge of ρ_1 and ρ_2 , such that $d_T(u_0, r_i) < d_T(v_0, r_i)$. Note that $\overrightarrow{v_0u_0}$ points downwards. Since subtrees T_i and T_j are independent, subtree T_j is contained in subtree $T_{u_0v_0}^{u_0} + u_0v_0$. According to Definition 4.1, to prove $x_i \in \text{polygon}(T_j) \subseteq \angle T_j$, it is sufficient to show that for any edge uv in $T_{u_0v_0}^{u_0} + u_0v_0$ with $d_T(v, v_0) < d_T(u, v_0)$, we have $x_i \in h_{vu}^v$.

Consider such an edge uv ; see Figure 16c. By Lemma 4.1, for $k = 1, 2$, we have $v_0, a_k, b_k \in h_{uv}^v$, and axis(uv) does not intersect the v_0 - b_k -path.

The u - v_0 -path cannot cross the v_0 - b_k -paths, since every greedy drawing is planar [ADF12]. Moreover, by Lemma 4.1, the u - v_0 -path lies in $h_{a_1b_1}^{a_1} \cap h_{a_2b_2}^{a_2} \subseteq \angle T_i$. Since all edges of $T_i - v_i r_i$ point upwards and $\overrightarrow{v_0u_0}$ points downwards, the u - v_0 -path lies in the part of $\angle T_i$ that is bounded by the b_1 - b_2 -path and contains u_0 , i.e., the u - v_0 -path cannot cross the gray area \diamond in Figure 16c bounded by the b_1 - b_2 -path and the rays ray (b_1, \vec{d}_1) and ray (b_2, \vec{d}_2) . Therefore, edge uv lies in $\angle T_i \setminus \diamond$.

Assume $x_i \notin h_{uv}^v$. Then, axis(uv) must intersect \diamond . Recall that axis(uv) cannot intersect the b_1 - b_2 -path. Then, axis(uv) must intersect a blue segment in Figure 16c. But then, edge uv cannot lie inside the cone $\angle T_i$, a contradiction.

Hence, $x_i \in h_{vu}^v$. Thus, $x_i \in \text{polygon}(T_j) \subseteq \angle T_j$.

(b) First, we prove $x_i = \text{apex}(T_i) \in \text{polygon}(T_j)$. Again, according to Definition 4.1, to prove $x_i \in \text{polygon}(T_j) \subseteq \angle T_j$, it is sufficient to show that for any edge uv chosen as in (a) we have $x_i \in h_{vu}^v$. The proof is identical to (a).

We now prove $\text{polygon}(T_j) \subseteq \nabla T_j$. Let a_1b_1, a_2b_2 be the extremal edges of T_j . For the cone $C_j = h_{a_1b_1}^{a_1} \cap h_{a_2b_2}^{a_2}$ from the definition of $\angle T_j$, we must have $r_j, v_j \in C_j$ by Lemma 4.1. Let $\overrightarrow{v_j r_j}$ point downward. Then, axis(a_1b_1) and axis(a_2b_2) are either parallel or intersect below axis($v_j r_j$); otherwise, it would be $|\angle T_j| > 0$.

Furthermore, since a_1b_1 and a_2b_2 are in $h_{v_j r_j}^{v_j}$ by Lemma 4.1, point p_1 is on axis(a_1b_1) above axis($v_j r_j$), and point p_2 is on axis(a_2b_2) above axis($v_j r_j$); see Figure 16a. Therefore, $C_j \cap h_{v_j r_j}^{v_j}$ (gray triangle) is contained in ∇T_j (orange triangle), and we have $\text{polygon}(T_j) \subseteq C_j \cap h_{v_j r_j}^{v_j} \subseteq \nabla T_j$. Since we have shown $x_i \in \text{polygon}(T_j)$, we have $x_i \in \nabla T_j$.

Finally, since p_1, p_2 lie on $T_j - v_j r_j$, we have: $p_1, p_2 \in \text{polygon}(T_i) \subseteq \angle T_i$.

(c) By Lemma 4.1, $T_j - v_j r_j$ lies in $\text{polygon}(T_i)$. Since p_1, p_2 lie on $T_j - v_j r_j$, we have $p_1, p_2 \in \text{polygon}(T_i)$. In (b), we have shown $\text{polygon}(T_i) \subseteq \nabla T_i$. Hence, $p_1, p_2 \in \nabla T_i$. \square

Lemma 4.6 (generalization of Claim 4 in [LM10]). *Let T_i, T_j be two independent subtrees of tree T . Consider a greedy drawing Γ of T . Then, either $|\angle T_i| > 0$ or $|\angle T_j| > 0$.*

Proof. Assume $|\angle T_i|, |\angle T_j| \leq 0$. By Lemma 4.5(c), ∇T_i contains the boundary segment $p_j^1 p_j^2$ of ∇T_j , and vice versa. This is not possible. \square

We shall use the following lemma to provide a certificate of non-existence of a greedy drawing.

Lemma 4.7. *Let $T_i, i = 1, \dots, d$ be pairwise independent subtrees of tree T . Consider a greedy drawing Γ of T . Then,*

$$\sum_{i=1, \dots, d, |\angle T_i| > 0} |\angle T_i| > (d-2)180^\circ.$$

Proof. First, let $\alpha_i = |\angle T_i| > 0, i = 1, \dots, d$. Arranging all angles $\angle T_i$ in accordance with Lemma 4.5(a) forms a convex $2d$ -gon, in which each $\angle T_i$ provides one angle of size α_i and the remaining d angles are less than 180° each. Then, $d \cdot 180^\circ + \sum_{i=1}^d \alpha_i > (2d-2)180^\circ$, and $\sum_{i=1}^d \alpha_i > (d-2)180^\circ$.

Now let $|\angle T_1| \leq 0$ and $\alpha_i = |\angle T_i| > 0, i = 2, \dots, d$. Then, arranging ∇T_1 and all the open angles $\angle T_i$ in accordance with Lemma 4.5(a) and (b) forms a convex $2d+1$ -gon, in which ∇T_1 provides two angles with sum 180° or less, each $\angle T_i, i = 2, \dots, d$ provides one angle of size α_i and the remaining d angles are less than 180° each. Then, $(d+1) \cdot 180^\circ + \sum_{i=2}^d \alpha_i > (2d-1)180^\circ$, and $\sum_{i=2}^d \alpha_i > (d-2)180^\circ$. \square

4.2.1 Example: independent stars

We now illustrate the above definitions using a simple example of independent stars. We consider rooted subtrees T_i that are stars, such that the root r_i is a leaf in T_i and v_i has degree 3, 4 or 5. Here, T_i does not need to contain all descendants of r_i that are in $T_{r_i v_i}^{v_i} + r_i v_i$. Consider a set of such stars that are all pairwise independent. For $k = 3, 4, 5$, let n_k be the number of stars with k leaves. Note that these stars do not necessarily cover all edges of T ; see Figure 17b.

Leighton and Moitra [LM10] showed that for $n_3 \geq 6$ no greedy drawing of T exists. We generalize this result slightly.

Lemma 4.8. *No greedy drawing of T exists if $n_3 + 2n_4 + 3n_5 \geq 6$.*

Proof. Consider a set of n_3 stars with three leaves, n_4 stars with four leaves and n_5 stars with five leaves, such that all stars are pairwise independent.

Recall that all angles in a greedy drawing are greater than 60° [ADF12]. Therefore, stars with three leaves have opening angle less than 120° . It follows that stars with four leaves have opening angle less than 60° . Furthermore, stars with five leaves cannot be drawn with an open angle. Thus, the sum of all positive opening angles is at most $120^\circ \cdot n_3 + 60^\circ \cdot n_4$.

From the lemma's assumption we have $-n_3 - 2n_4 - 3n_5 \leq -6$. By adding $3n_3 + 3n_4 + 3n_5$ to both sides, we acquire $2n_3 + n_4 \leq 3n_3 + 3n_4 + 3n_5 - 6$. Finally, multiplying by 60° provides $120^\circ \cdot n_3 + 60^\circ \cdot n_4 \leq (n_3 + n_4 + n_5 - 2)180^\circ$, a contradiction to Lemma 4.7. \square

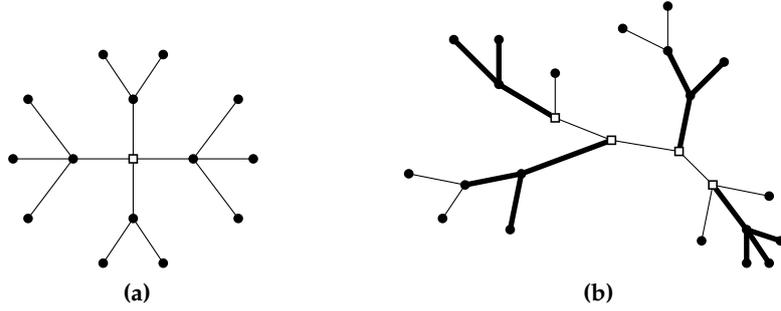


Figure 17: Independent stars with roots drawn as hollow squares. (a) A tree formed by two stars with three leaves and two stars with four leaves, all of which are pairwise independent. Since $1 \cdot 2 + 2 \cdot 2 + 3 \cdot 0 = 6$, by Lemma 4.8, the tree has no greedy drawing. (b) A tree containing three stars with three leaves and one star with four leaves, all of which are pairwise independent (thick). Although $1 \cdot 3 + 2 \cdot 1 < 6$, we shall see later that the tree has no greedy drawing.

Lemma 4.8 is illustrated in Figure 17.

4.2.2 Outline of the characterization

Consider a node $r \in V$ with neighbors v_1, \dots, v_d . The subtrees $T_i = T_{r v_i}^{v_i} + r v_i$ with the common root r are pairwise independent, $i = 1, \dots, d$. Consider angles $\varphi_i \geq 0$, such that $|\angle T_i| \leq \varphi_i$ in any greedy drawing of T_i . If either there exist $i, j \in \{1, \dots, d\}$, $i \neq j$, such that $\varphi_i, \varphi_j = 0$, or $\sum_{i=1}^d \varphi_i \leq (d - 2)180^\circ$, then, by Lemmas 4.6 and 4.7, the tree T has no greedy drawing.

Determining tight upper bounds on $|\angle T_i|$ will let us derive a sufficient condition for the existence of a greedy drawing of T . Using the next result, we shall be able to compute such a bound for any rooted subtree in Section 4.3.

4.2.3 Shrinking lemma

In this section, we present the *shrinking lemma* which is crucial for later proofs.

Again, consider a greedy drawing Γ of tree T and subtree T_i of T with root r_i . Let v_i be the only neighbor of r_i in T_i . First, we show that replacing T_i by a segment connecting r_i to $\text{apex}(T_i)$ keeps the drawing greedy.

Lemma 4.9. *Let $T_i = (V_i, E_i)$, $T_i = T_{r_i v_i}^{v_i} + r_i v_i$, be drawn with an open angle. Consider a new vertex $x_i \notin V$ and the tree $T' = (T - (V_i \setminus \{r_i\})) + r_i x_i$. The drawing Γ' of T' created by 1) assigning the coordinates of $\text{apex}(T_i)$ to vertex x_i , 2) removing edges E_i from Γ and 3) adding the segment $r_i x_i$, is greedy.*

Furthermore, the circular order of edges incident to r_i is the same in Γ and Γ' , apart from $r_i x_i$ replacing $r_i v_i$.

Proof. First, let $\text{apex}(T_i)$ be a vertex of T_i like in the second case of Definition 4.4. We remove all edges of T_i that are not on the path from r_i to $\text{apex}(T_i)$. Note

that the resulting drawing is greedy. Then, by repeatedly applying Lemma 4.4 to intermediate vertices of that path we see that the resulting drawing Γ' of T' is greedy.

Now assume that the first case of Definition 4.4 holds. To prove that Γ' is greedy, by Lemma 4.2, it is sufficient to show that

- (1) for every edge uv of $T - (V_i \setminus \{r_i\})$ with $d_T(v, r_i) < d_T(u, r_i)$, the new vertex x_i is in h_{uv}^v and that
- (2) every vertex w of $T - (V_i \setminus \{r_i\})$ is in $h_{r_i x_i}^{r_i}$.

The proof of $x_i \in h_{uv}^v$ is identical to the corresponding part of the proof of Lemma 4.5(a).

We now consider the second statement. Let w be a vertex in $T - (V_i \setminus \{r_i\})$. For $k = 1, 2$, let ρ_k be the r_i - b_k -path in T_i and $u_0 v_0$ the last common edge of ρ_1 and ρ_2 , such that $d_T(u_0, r_i) < d_T(v_0, r_i)$. Note that v_0 lies on the b_k - w -path. Therefore, we have $w \in \Lambda = h_{v_0 b_1}^{v_0} \cap h_{v_0 b_2}^{v_0}$.

As in the proof of Lemma 4.5(a), without loss of generality, we can assume that $\overrightarrow{a_1 b_1}$ and $\overrightarrow{a_2 b_2}$ point upwards and $\overrightarrow{v_i r_i}$ points downwards. Then, all edges of $T_i - v_i r_i$ point upwards.

Consider the convex quadrilateral \diamond_1 bounded by the cone $\angle T_i$ and the rays $\text{ray}(v_0, \overrightarrow{a_1 b_1})$ and $\text{ray}(v_0, \overrightarrow{a_2 b_2})$; see Figure 18a. Since edges $a_1 b_1$ and $a_2 b_2$ are extremal, paths ρ_1 and ρ_2 must lie in \diamond_1 . Therefore, the convex quadrilateral $\diamond_2 = v_0 b_1 x_i b_2$ is contained in \diamond_1 , and the angles of \diamond_2 at b_1 and b_2 are at least 90° . Then, by Lemma 4.10, we have $\Lambda \subseteq h_{v_0 x_i}^{v_0}$, and vertex w is in $h_{v_0 x_i}^{v_0}$. Additionally, we have $|wr_i| < |wv_0|$. Thus, $|wr_i| < |wv_0| < |wx_i|$, and $w \in h_{r_i x_i}^{r_i}$.

Finally, consider the clockwise circular order of edges incident to r_i in Γ . Let edge $r_i w_1$ be the predecessor of $r_i v_i$ and let $r_i w_2$ be the successor of $r_i v_i$. Let Λ' be the cone with boundary rays $\text{ray}(r_i, \overrightarrow{r_i w_1})$ and $\text{ray}(r_i, \overrightarrow{r_i w_2})$, such that $v_i \in \Lambda'$. To show the last statement of the lemma, it is sufficient to show $x_i \in \Lambda'$. We have shown $x_i \in h_{r_i w_1}^{r_i} \cap h_{r_i w_2}^{r_i} \cap h_{r_i v_i}^{v_i}$. Since we have $h_{r_i w_1}^{r_i} \cap h_{r_i w_2}^{r_i} \cap h_{r_i v_i}^{v_i} \subseteq \Lambda'$ (see Figure 18c), the statement of the lemma follows. \square

Lemma 4.10. *Consider a convex quadrilateral with vertices a, b, c, d in this order along the boundary. Let $\angle abc \geq 90^\circ$ and $\angle adc \geq 90^\circ$. Then, $h_{ab}^a \cap h_{ad}^a \subseteq h_{ac}^a$.*

Proof. Without loss of generality, let \overrightarrow{ac} be vertical and point upwards, let \overrightarrow{ab} point upwards and to the left and \overrightarrow{ad} point upwards and to the right; see Figure 18b. Then, $\text{axis}(ac)$ is horizontal, and both boundary rays of the cone $\Lambda = h_{ab}^a \cap h_{ad}^a$ point downwards. Therefore, to prove $\Lambda \subseteq h_{ac}^a$, it is sufficient to show that the apex x of Λ lies on $\text{axis}(ac)$ or below it.

Consider the circle C with center x and radius $|ax|$. Since x is the intersection of $\text{axis}(ab)$ and $\text{axis}(ad)$, points b and d also lie on C . Since $\angle abc \geq 90^\circ$ and $\angle adc \geq 90^\circ$, point c cannot lie in the interior of C , and we have $|ax| \leq |xc|$. Therefore, x lies on $\text{axis}(ac)$ or below it, and we have $\Lambda \subseteq h_{ac}^a$. \square

We now prove the main result of this section.

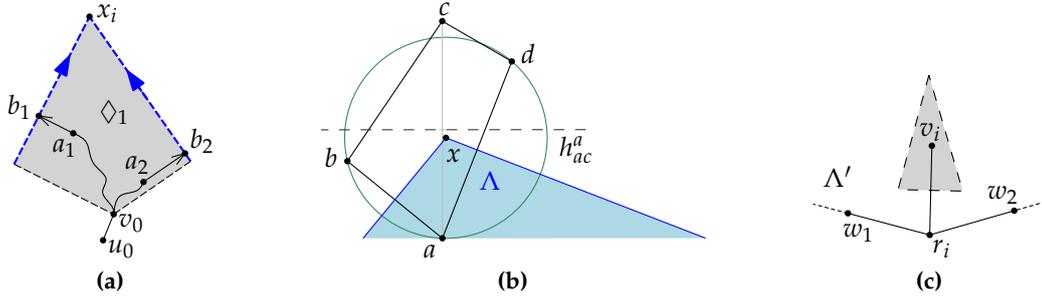


Figure 18: (a) Convex quadrilateral \diamond_1 from the proof of Lemma 4.9. (b) Proof of Lemma 4.10. Point c does not lie inside the circle through points a, b, d . Hence, $\Lambda = h_{ab}^a \cap h_{ad}^a \subseteq h_{ac}^a$. (c) Point x_i lies inside the gray area $h_{r_i w_1}^{r_i} \cap h_{r_i v_i}^{r_i}$.

Lemma 4.11 (Shrinking lemma). *Let $T_i = (V_i, E_i)$, $T_i = T_{r_i v_i}^{v_i} + r_i v_i$, be drawn with an open angle, and let $x_i = \text{apex}(T_i)$. There exists $\varepsilon_M > 0$, such that for every $0 < \varepsilon \leq \varepsilon_M$, there exists a greedy drawing Γ_ε of T created by applying a translation and a scaling to the vertices $V_i \setminus \{r_i\}$, such that every transformed vertex from $V_i \setminus \{r_i\}$ has distance at most ε to x_i . Furthermore, the circular order of edges incident to r_i is the same in Γ and Γ_ε .*

Proof. If T_i has only one edge, $\Gamma_\varepsilon = \Gamma$ satisfies the lemma statement. Now let T_i have more than one edge.

Let C be the cone $\angle T_i$, and let $x_i = \text{apex}(T_i)$. Since $\text{polygon}(T_i) \subseteq C$, cone C contains all vertices $V \setminus (V_i \setminus \{r_i\})$ in its interior; see Figure 19a.

Consider the drawing Γ' of $T' = (T - (V_i \setminus \{r_i\})) + r_i x_i$ which is greedy by Lemma 4.9. The circular order of edges incident to r_i is the same in Γ and Γ' . Let T' have root x_i . Then, by Lemma 4.1, we have $x_i \in \text{polygon}(T')$. Then, there exists $\varepsilon_M > 0$, such that the circle with radius ε_M and center x_i is contained in $\text{polygon}(T')$. Such a circle is shown in Figure 19b.

Next, consider vertices $V_i \setminus \{r_i\}$ together with their coordinates in the original drawing Γ . Again, let v_i be the neighbor of r_i in T_i . For $0 < \varepsilon < \varepsilon_M$, we apply a scaling to $V_i \setminus \{r_i\}$, such that all vertices in $V_i \setminus \{r_i\}$ have distance at most ε from v_i . Next, we apply a translation, such that vertex v_i is moved to the point x_i . Let Γ_ε be the resulting drawing. We now show that Γ_ε is greedy. By construction of Γ_ε , vertices $V_i \setminus \{r_i\}$ lie in $\text{polygon}(T')$.

The shrunken drawing of $T_i - r_i$ is greedy. It remains to be shown that for every edge uv of $T_i - r_i$ with $d_T(u, r_i) < d_T(v, r_i)$, we have $V \setminus (V_i \setminus \{r_i\}) \subseteq h_{uv}^u$, i.e., the axis of every shrunken edge of $T_i - r_i$ crosses none of the old edges $E \setminus E_i$. Since edges $a_1 b_1$ and $a_2 b_2$ are extremal edges of T_i , all edges of $T_i - r_i$ lie in the cone with apex x_i and boundary rays $\text{ray}(x_i, \overrightarrow{a_1 b_1})$ and $\text{ray}(x_i, \overrightarrow{a_2 b_2})$; see the gray area in Figure 19c. Moreover, since $\angle_{\text{cw}}(a_1 b_1, \overrightarrow{uv}) \leq \angle_{\text{cw}}(a_1 b_1, a_2 b_2)$, the cone C is contained in h_{uv}^u . Since $V \setminus (V_i \setminus \{r_i\}) \subseteq C$, we have $V \setminus (V_i \setminus \{r_i\}) \subseteq h_{uv}^u$. \square

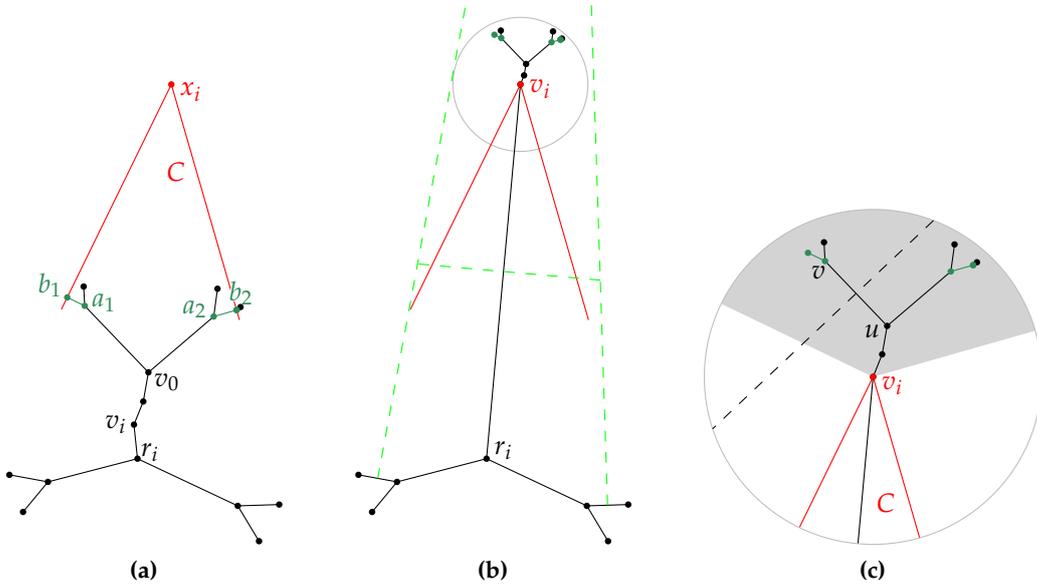


Figure 19: Illustration of Lemma 4.11. (a) Greedy drawing Γ of T . Cone $C = \angle T_i$ is drawn red. (b) Greedy drawing Γ' of T created by applying the transformation in Lemma 4.11. Note that the normals of the edges of $T_{r_i v_i}^i + r_i v_i$ do not cross the circle. (c) We have $C \subseteq h_{uv}^u$; see the proof of Lemma 4.11.

4.3 OPENING ANGLES OF ROOTED TREES

The main idea of our decision algorithm is to process the nodes of T bottom-up while calculating tight upper bounds on the maximum possible opening angles of the considered subtrees. If T contains a node of degree 5, it cannot be drawn with an open angle, since each pair of consecutive edges forms an angle strictly greater than 60° . In this section, we consider trees with maximum degree 4.

If a subtree T' can be drawn with an open angle $\varphi - \varepsilon$ for any $\varepsilon > 0$, but not φ , we say that it has opening angle φ^- and write $|\angle T'| = \varphi^-$. For example, a triple has opening angle 120°^- and a quadruple 60°^- . We call a subtree *non-trivial* if it is not a single node or a simple path. Figure 20 shows possibilities to combine or extend non-trivial subtrees T', T_1, T_2, T_3 . Let r be the root of subtrees T', T_1, T_2, T_3 and r_0 the root of the combined subtree \bar{T} .

We shall now prove tight bounds on the possible opening angles for each construction. As we show later, only cases I–V are feasible for the resulting subtree to have an open angle.

Lemma 4.12. *Let T' be a subtree with positive opening angle. Consider the subtree $\bar{T} = T' + rr_0$ in Figure 20a. Then \bar{T} has the same maximum opening angle as T' .*

Proof. Obviously, subtree \bar{T} cannot have a bigger maximum opening angle than T' . By Lemma 4.11, for every greedy drawing of T' there exists a greedy drawing with an opening angle $\angle T'$ of the same size in which $T_{r_0}^{r'}$ is drawn arbitrarily small. We then draw v, r, r_0 collinearly inside $\angle T'$. \square

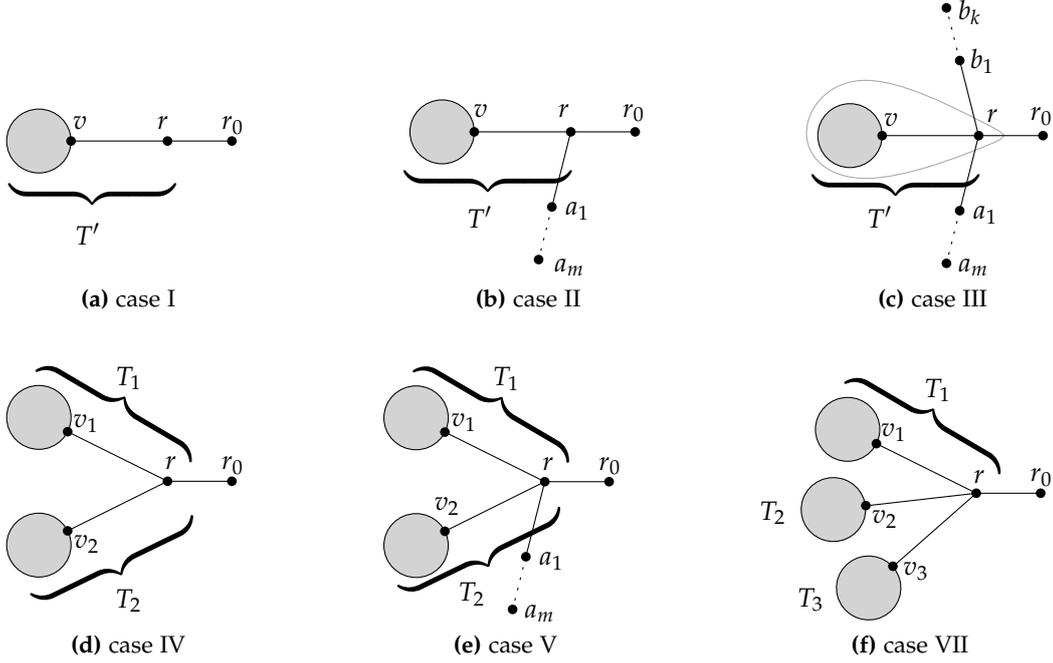


Figure 20: (a)–(e): Possible cases when combining subtrees to maintain an open angle. Subtrees T_1, T_2 have opening angles $\in (90^\circ, 120^\circ)$. In case VII ((f)) or in case VI ($|\angle T_i| \leq 90^\circ$ in IV or V for one $i \in \{1, 2\}$) no open angle is possible.

To compute the maximum opening angle of the combined subtree \bar{T} in cases II–V, we use the following strategy. We show that applying Lemma 4.11 to T' does not decrease the opening angle of \bar{T} in a drawing. Hence, it suffices to consider only drawings in which T'_{rv} is shrunk sufficiently. We then obtain an upper bound by solving a linear maximization problem. Finally, we construct a drawing with an almost-optimal opening angle for \bar{T} inductively using an almost-optimal construction for T' . Tight upper bounds on opening angles of the combined subtree \bar{T} for all possible cases are listed in Table 2. Note that no bounds in $(120^\circ, 180^\circ)$ and $(60^\circ, 90^\circ]$ appear. We now present the proofs for cases II–V.

For one of the boundary rays of $\angle \bar{T}$ and an edge uv of \bar{T} , we say that uv induces the corresponding boundary if uv is extremal in \bar{T} and $\text{axis}(uv)$ is parallel to that boundary.

Lemma 4.13. *Let T' be a subtree with $\angle T' = \varphi^-$, and consider the subtree $\bar{T} = T' + rr_0 + ra_1 + a_1a_2 + \dots + a_{m-1}a_m$ in Figure 20b. Then,*

- (i) $|\angle \bar{T}| = (45^\circ + \frac{\varphi}{2})^-$ if $\varphi > 90^\circ$, and
- (ii) $|\angle \bar{T}| = \varphi^-$ if $\varphi \leq 90^\circ$.

Proof. First, let $m = 1$.

- (i) Consider a greedy drawing Γ of \bar{T} . Let ra_1 be drawn horizontally, such that $\overrightarrow{ra_1}$ points to the right, and let v be above it and to the left of $\text{axis}(ra_1)$; see Figure 21a, b, d.

Table 2: Computing maximum opening angle of the combined subtree \bar{T} . Let $|\angle T_i| = \varphi_i^-$, $\varphi_i \geq \varphi_{i+1}$, and $|\angle T_i| = \varphi_i = 180^\circ$ if T_i is a path.

case	φ_1	φ_2	φ_3	maximum $ \angle \bar{T} $	proof
I	$(0^\circ, 180^\circ]$	-	-	φ_1	Lem. 4.12
II.i	180°	$(90^\circ, 120^\circ]$	-	$(\frac{\varphi_2}{2} + 45^\circ)^-$ $\in (90^\circ, 120^\circ)$	Lem. 4.13
II.ii	180°	$(0^\circ, 60^\circ]$	-	$\varphi_2^- \in (0^\circ, 60^\circ)$	Lem. 4.13
III	180°	180°	$(0^\circ, 120^\circ]$	$\frac{\varphi_3}{2}^- \in (0^\circ, 60^\circ)$	Lem. 4.14
IV	$(90^\circ, 120^\circ]$	$(90^\circ, 120^\circ]$	-	$(\varphi_1 + \varphi_2 - 180^\circ)^-$ $\in (0^\circ, 60^\circ)$	Lem. 4.15
V	180°	$(90^\circ, 120^\circ]$	$(90^\circ, 120^\circ]$	$(\frac{3}{4}\varphi_2 + \frac{\varphi_3}{2} - 112.5^\circ)^-$ $\in (0^\circ, 60^\circ)$	Lem. 4.16
VI	$(0^\circ, 120^\circ]$	$(0^\circ, 90^\circ]$	-	$< 0^\circ$	Lem. 4.17
VII	$(0^\circ, 120^\circ]$	$(0^\circ, 120^\circ]$	$(0^\circ, 120^\circ]$	$< 0^\circ$	Lem. 4.17

First, we show $|\angle \bar{T}| < 45^\circ + \frac{\varphi}{2}$. Let vector \vec{rv} point upwards and to the right; otherwise, $|\angle \bar{T}| \leq 90^\circ$. Due to Lemma 4.3, the right boundary of $\angle \bar{T}$ is induced by ra_1 . The left boundary either (1) coincides with the left boundary of $\angle T'$ (see Figure 21a), or (2) is induced by rv (Figure 21b). We apply Lemma 4.11 to T'_{rv} in Γ and acquire Γ' , in which T'_{rv} is drawn arbitrarily small. In Γ' , edge ra_1 still induces the right boundary of $\angle \bar{T}$. In case (1), the left boundary of $\angle \bar{T}$ is again formed by the left boundary of $\angle T'$, and $|\angle \bar{T}|$ remains the same. In case (2), the subtree T'_{rv} must lie to the right of the line through rv in Γ (since for each edge uw in T'_{rv} with $\text{dist}(r, u) < \text{dist}(r, w)$, we have $\angle_{\text{cw}}(\vec{uw}, \vec{ra}_1) \leq \angle_{\text{cw}}(\vec{rv}, \vec{ra}_1)$), and so does the point $x = \text{apex}(T')$. Thus, the edge rv is turned clockwise in Γ' , and $|\angle \bar{T}|$ increases; see Figure 21c. Therefore, to acquire an upper bound for $|\angle \bar{T}|$ it suffices to only consider drawings in which T'_{rv} is drawn arbitrarily small. Let $\alpha = \angle a_1rv$. Then, for $\bar{\varphi} = |\angle \bar{T}|$ it holds: $\bar{\varphi} \leq 180^\circ - \alpha$, $\bar{\varphi} < \varphi - 90^\circ + \alpha$; see the blue and green angles in Figure 21d. Thus, $\bar{\varphi}$ lies on the graph of the function $f(\alpha) = 180 - \alpha$ or below it and strictly below the graph of the function $g(\alpha) = \varphi - 90 + \alpha$. Maximizing over α gives $\bar{\varphi} < 45 + \frac{\varphi}{2}$. We can achieve $\bar{\varphi} = (45^\circ + \frac{\varphi}{2})^-$ by choosing $\alpha = 135^\circ - \frac{\varphi}{2} + \varepsilon'$ and drawing T'_{rv} sufficiently small with $|\angle T'| = \varphi - \varepsilon$ for sufficiently small $\varepsilon, \varepsilon' > 0$.

(ii) Obviously, $|\angle T'| \geq |\angle \bar{T}|$. For the second part, see Figure 21e. We now choose $\angle a_1rv = 90^\circ - \frac{\varepsilon}{2}$ and draw ra_1 long enough, such that its axis does not cross T'_{rv} . We rotate T'_{rv} such that the right side of $\angle T'$ and edge rv form an angle $\frac{3\varepsilon}{2}$. Then, the left boundary of $\angle \bar{T}$ is defined by the left boundary of $\angle T'$, and the right

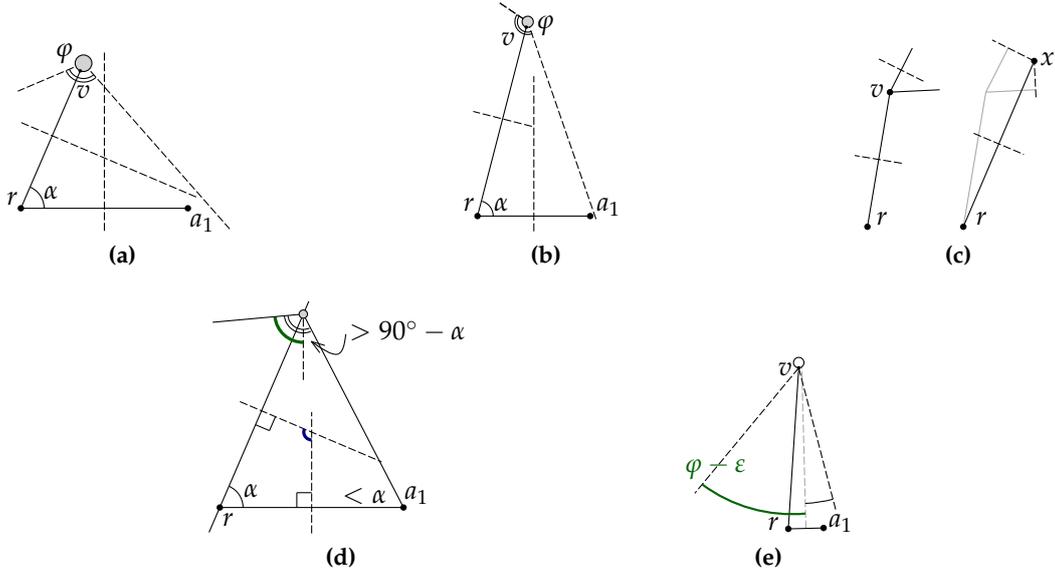


Figure 21: Optimal construction and tight upper bound for case II; see Lemma 4.13.

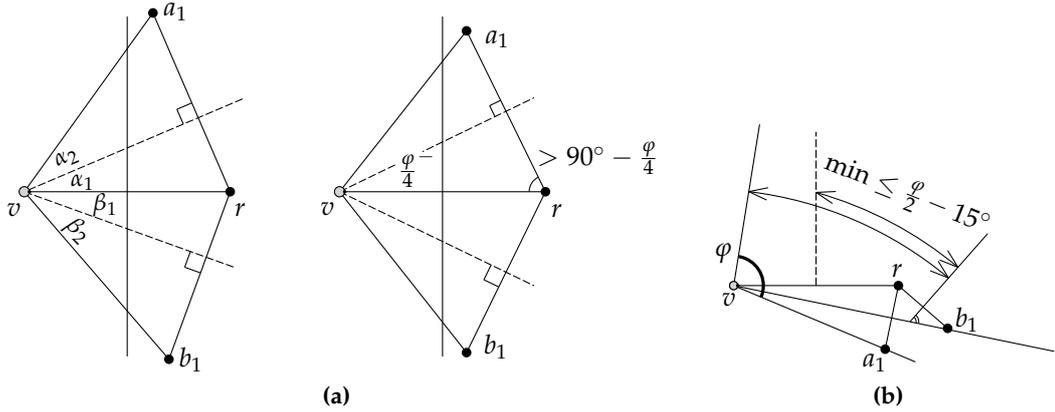


Figure 22: Optimal construction and tight upper bound for case III; see Lemma 4.14.

boundary of $\angle \bar{T}$ is induced by the edge ra_1 . Therefore, the opening angle of \bar{T} in the drawing is $\varphi - \varepsilon$.

For $m \geq 2$, draw a_2, \dots, a_m collinear with ra_1 and arbitrarily close to a_1 . □

Lemma 4.14. *Let T' be a subtree with $|\angle T'| = \varphi^- < 120^\circ$, and consider subtree $\bar{T} = T' + rr_0 + ra_1 + \dots + a_{m-1}a_m + rb_1 + \dots + b_{k-1}b_k$ in Figure 20c. Then, $\angle \bar{T} = \frac{\varphi^-}{2}$.*

Proof. First, let $k = m = 1$. Consider a greedy drawing Γ of \bar{T} with $|\angle \bar{T}| > 0$. Let rv be horizontal in Γ and let v lie to the left of r . There exist two possibilities for Γ . The edge rv can be either drawn inside the angle $\angle a_1rb_1 < 180^\circ$ (see Figure 22a) or on the outside of it (see Figure 22b).

In the first case, let a_1 lie above rv and b_1 below. Then, the upper boundary of $\angle \bar{T}$ is induced by ra_1 and the lower by rb_1 . This remains the case after

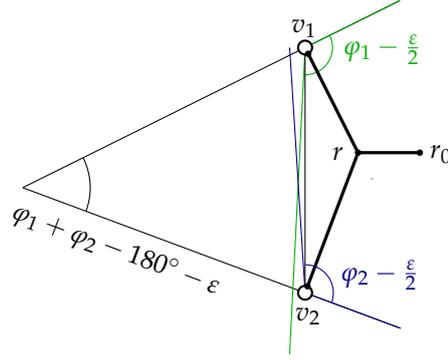


Figure 23: Sketch of the proof of Lemma 4.15.

applying Lemma 4.11 to T' ; see Figure 22a for the corresponding drawing Γ' . We can assume that in Γ' , we have $\angle vra_1, \angle vrb_1 < 90^\circ$ (otherwise, we can increase $\angle \bar{T}$ by turning ra_1 counterclockwise or rb_1 clockwise). By greediness, we must have $\alpha_1 < \alpha_2, \beta_1 < \beta_2$ and $\alpha_1 + \alpha_2 + \beta_1 + \beta_2 < \varphi$. Thus, the opening angle in this construction is $\alpha_1 + \beta_1 < \frac{\varphi}{2}$. The angle $|\angle \bar{T}| = \frac{\varphi}{2}^-$ can be achieved by choosing $\alpha_1 = \beta_1 = \frac{\varphi}{4} - 2\varepsilon, \alpha_2 = \beta_2 = \frac{\varphi}{4} - \varepsilon$ for a sufficiently small ε . Then, $\angle a_1vr < \angle a_1rv$, and $|ra_1| < |a_1v|$ (we have $\frac{\varphi}{2}^- < 60^\circ \leq 90^\circ - \frac{\varphi}{4}$). Hence, the drawing is greedy and has opening angle $|\angle \bar{T}| = \frac{\varphi}{2}^-$.

Now consider the second option for Γ ; see Figure 22b. Let ra_1 be inside $\angle vrb_1 < 180^\circ$. Then, rb_1 induces a boundary of the opening angle of \bar{T} and ra_1 induces a boundary of the opening angle of $\bar{T} - rb_1$. By Lemma 4.13, $|\angle(\bar{T} - rb_1)| < \frac{\varphi}{2} + 45^\circ$, and, since $\angle a_1rb_1 > 60^\circ$, the drawing of \bar{T} has opening angle at most $\frac{\varphi}{2} + 45^\circ - 60^\circ = \frac{\varphi}{2} - 15^\circ$. Hence, the first option is optimal. We add a_2, \dots, a_m and b_2, \dots, b_k similarly to the proof of Lemma 4.13.

Note that the new opening angle is $< 60^\circ$. \square

Lemma 4.15. *Let T_1, T_2 be subtrees with $|\angle T_i| = \varphi_i^- \in (90^\circ, 120^\circ), i = 1, 2$, and consider subtree $\bar{T} = T_1 + T_2 + rr_0$ in Figure 20d. Then, $\angle \bar{T} = (\varphi_1 + \varphi_2 - 180^\circ)^-$.*

Proof. Let r_0, v_1 and v_2 be ordered counterclockwise around r and let v_1 be above v_2 in Γ . Then, the upper boundary of $\angle \bar{T}$ is formed either by the upper boundary of $\angle T_1$ or induced by rr_0 , and the lower boundary of $\angle \bar{T}$ is formed either by the lower boundary of $\angle T_2$ or induced by rr_0 . Since $\text{apex}(T_1) \in \angle T_2$ and $\text{apex}(T_2) \in \angle T_1$, an opening angle $|\angle \bar{T}| \geq \varphi_1 + \varphi_2 - 180^\circ$ is not possible.

For the lower bound, see the construction in Figure 23. Both $(T_i)_{rv_i}^{v_i}$ are drawn arbitrarily small. The lower boundary ray of $\angle T_1$ and the upper boundary ray of $\angle T_2$ have intersection angle ε , and the other two sides form an angle $\varphi' = \varphi_1 + \varphi_2 - 180^\circ - \varepsilon$. The edges rv_1 and rv_2 are drawn orthogonal to the upper boundary of $\angle T_1$ and lower boundary of $\angle T_2$ respectively, so their axes are parallel to the boundary rays of $\angle \bar{T}$. We have $\angle v_1rv_2 = 360^\circ - \varphi_1 - \varphi_2 + \varepsilon \geq 120^\circ + \varepsilon$. Hence, no axis crosses another edge, and the drawing is greedy. Note that the new opening angle is $< 60^\circ$. \square

formed by the upper boundary of $\angle T_1$ and $\overrightarrow{v_1 v_2}$. We must have $\alpha < \alpha_1$ and $\gamma < \varphi_1$. Then, for $\varphi' = |\angle \overline{T}|$ we must have:

$$\begin{aligned} \varphi' &< (90^\circ - \alpha_1) + \frac{\varphi_2 - \alpha}{2} < 90^\circ - \frac{3}{2}\alpha + \frac{\varphi_2}{2} =: f(\alpha), \\ \varphi' &< (\alpha + \varphi_1 - 180^\circ) + \frac{\varphi_2 - \alpha}{2} = \frac{\alpha}{2} - 180^\circ + \varphi_1 + \frac{\varphi_2}{2} =: g(\alpha). \end{aligned}$$

Hence, $\varphi' < \varphi_{\max} := \max_{\alpha} \min\{f(\alpha), g(\alpha)\} = \frac{3}{4}\varphi_1 + \frac{1}{2}\varphi_2 - 112.5^\circ$. We can achieve φ_{\max} by choosing $\alpha = 135^\circ - \frac{\varphi_1}{2}$, $\alpha_1 = \alpha + \varepsilon$, $\gamma = \varphi_1 - \varepsilon$, $\angle v_1 v_2 a_1 = \varphi_2 - \varepsilon$ and $\beta = 157.5^\circ - \frac{\varphi_1}{4} - \frac{\varphi_2}{2} + \varepsilon$. In this construction, $\text{axis}(rv_1)$ is parallel to the upper boundary ray of $\angle T_1$ (dashed green in Figure 24b).

Since we assumed $\varphi_1 > 90^\circ$, we have $\frac{\varphi_1}{4} > 22.5^\circ$, and the second embedding option provides a bigger opening angle. Note that the new opening angle is $< 37.5^\circ$. \square

Lemma 4.17. *If either (VI) $|\angle T_i| < 90^\circ$ in Figure 20d or 20e for some $i \in \{1, 2\}$ or (VII) $|\angle T_i| < 120^\circ$ for each $i = 1, \dots, 3$ in Figure 20f, we have $|\angle \overline{T}| < 0$.*

Proof. First, let $|\angle T_1| < 120^\circ$ and $|\angle T_2| \leq 90^\circ$ in Figure 20d or 20e. Since no tight upper bounds in range $(60^\circ, 90^\circ]$ appear (see Table 2), we have $|\angle T_2| < 60^\circ$. We must have $\text{apex}(T_i) \in \angle T_j$ for $i \neq j$, therefore, no open angle is possible; see Figure 25a. The same holds for $|\angle T_1|, |\angle T_2|, |\angle T_3| < 120^\circ$; see Figure 25b. \square

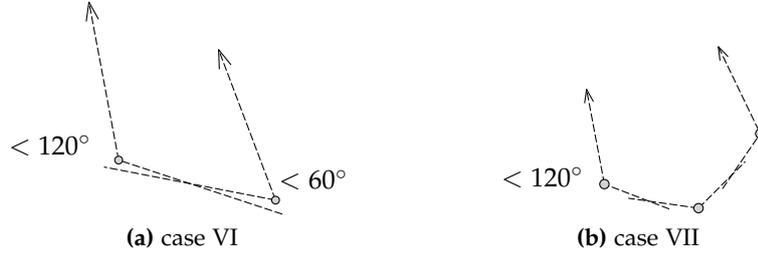


Figure 25: In cases VI and VII, no open angle is possible.

4.4 ARRANGING ROOTED SUBTREES WITH OPEN ANGLES

In this section, we consider the task of constructing a greedy drawing Γ of T by combining independent rooted subtrees with a common root. The following problem (restricted to $n \in \{3, 4, 5\}$) turns out to be fundamental in this context.

Problem 1. *Given n angles $\varphi_0, \dots, \varphi_{n-1} > 0^\circ$, is there a convex n -gon P with corners v_0, \dots, v_{n-1} (in arbitrary order) with interior angles $\psi_i < \varphi_i$ for $i = 0, \dots, n - 1$, such that the star $K_{1,n}$ has a greedy drawing with root r inside P and leaves v_0, \dots, v_{n-1} ?*

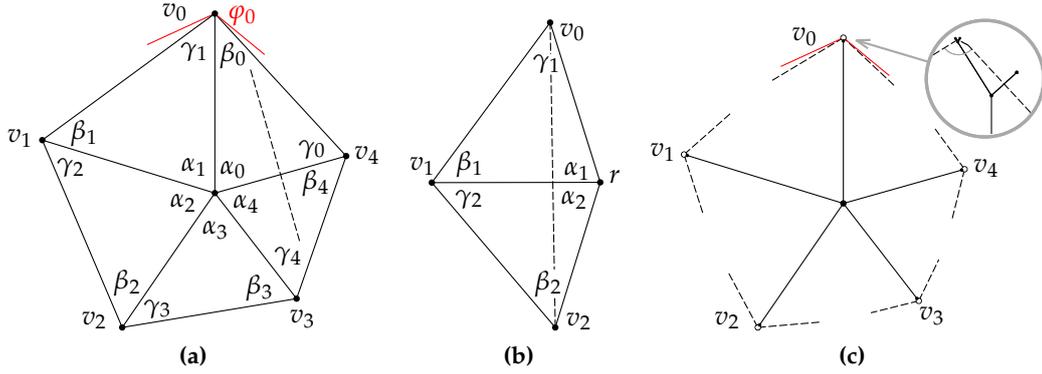


Figure 26: (a) Sketch for the optimization problem (*). (b) In a drawing of $K_{1,n}$ induced by a solution of (*), path v_i, r, v_j is distance-decreasing. (c) Solving (*) lets us construct greedy drawings by placing sufficiently small drawings of subtrees at n -gon corners.

If Problem 1 has a solution we write $\{\varphi_0, \dots, \varphi_{n-1}\} \in \mathcal{P}^n$. Note that $\{\varphi_0, \dots, \varphi_{n-1}\} \in \mathcal{P}^n$ is a multiset. Problem 1 can be solved using a series of the following optimization problems (one for each fixed cyclic ordering of $(\varphi_1, \dots, \varphi_n)$).

$$\begin{aligned}
 & \text{maximize } \varepsilon \text{ under: } \varepsilon, \alpha_i, \beta_i, \gamma_i \in [0^\circ, 180^\circ], i = 0, \dots, n-1 \\
 & \beta_i + \varepsilon \leq \alpha_i, \quad \gamma_i + \varepsilon \leq \alpha_i, \quad \beta_i + \gamma_{i+1} + \varepsilon \leq \varphi_i \quad (i \bmod n) \\
 & \alpha_i + \beta_i + \gamma_i = 180^\circ, \quad \alpha_0 + \dots + \alpha_{n-1} = 360^\circ \\
 & \sin(\beta_0) \cdot \dots \cdot \sin(\beta_{n-1}) = \sin(\gamma_0) \cdot \dots \cdot \sin(\gamma_{n-1})
 \end{aligned} \tag{*}$$

The last constraint in (*) follows from applying the law of sines and is known as the wheel condition in the work of Di Battista and Vismara [DV96].

Lemma 4.18. *We have $\{\varphi_0, \dots, \varphi_{n-1}\} \in \mathcal{P}^n$ if and only if there exists a solution of (*) with $\varepsilon > 0$ for an ordering $(\varphi_0, \dots, \varphi_{n-1})$.*

Proof. A solution to Problem 1 provides a solution to (*) by the construction in Figure 26a, since $|rv_{i-1}|, |rv_i| < |v_{i-1}v_i| \Leftrightarrow \beta_i, \gamma_i < \alpha_i$.

Conversely, a solution to (*) with $\varepsilon > 0$ provides a greedy drawing of $K_{1,n}$ as follows. We use angles $\alpha_i, \beta_i, \gamma_i$ from the solution of (*) to construct a drawing of a $K_{1,n}$ inside a convex n -gon as shown in Figure 26a. To show that the drawing is greedy, consider leaves s, t of $K_{1,n}$. First, let $s = v_{i-1}$ and $t = v_i$. Since $\angle v_{i-1}rv_i = \alpha_i > \gamma_i$, we have $|rt| < |st|$. Now let $s = v_{i-2}$, $t = v_i$ and $n = 4, 5$. If $n = 5$, then $\angle_{\text{ccw}}(\overrightarrow{rv_{i-2}}, \overrightarrow{rv_i}) \leq 180^\circ$, since all α_j are at least 60° . For $n = 4$, we have either $\angle_{\text{ccw}}(\overrightarrow{rv_{i-2}}, \overrightarrow{rv_i}) \leq 180^\circ$ or $\angle_{\text{cw}}(\overrightarrow{rv_{i-2}}, \overrightarrow{rv_i}) \leq 180^\circ$. Without loss of generality, assume $\angle_{\text{ccw}}(\overrightarrow{rv_{i-2}}, \overrightarrow{rv_i}) \leq 180^\circ$, and consider $s = v_0$ and $t = v_2$; see Figure 26b. We have $\alpha_1 + \alpha_2 \geq 120^\circ$, hence, $|rv_0|, |rv_2| < |v_0v_2|$. For the remaining choices of s, t , the proof is analogous. \square

Deciding whether a solution of (*) with $\varepsilon > 0$ exists is in fact equivalent to deciding whether the wheel condition can be satisfied in the interior of a $2n - 1$ -dimensional polytope.

Observation 4.3. *Let $n \in \{3, 4, 5\}$ and $\varphi_i \in [0^\circ, 180^\circ]$ for $i = 0, \dots, n - 1$, such that*

$$\sum_{i=0}^{n-1} \varphi_i > (n - 2)180^\circ.$$

For a permutation τ of $(0, \dots, n - 1)$, define a $2n - 1$ -dimensional polytope P_τ as follows:

$$P_\tau = \left\{ (\beta_0, \dots, \beta_{n-1}, \gamma_0, \dots, \gamma_{n-1}) \left| \begin{array}{l} \text{for } i = 0, \dots, n - 1 : \\ \beta_i \geq 0, \gamma_i \geq 0, \\ \beta_i + \gamma_{i+1} \leq \varphi_{\tau(i)}, \\ 180^\circ - \beta_i - \gamma_i \geq \beta_i, \\ 180^\circ - \beta_i - \gamma_i \geq \gamma_i; \\ \sum_{i=0}^{n-1} (\beta_i + \gamma_i) = (n - 2)180^\circ \end{array} \right. \right\}.$$

Define

$$\omega(\beta_0, \dots, \beta_{n-1}, \gamma_0, \dots, \gamma_{n-1}) = \prod_{i=0}^{n-1} \sin(\beta_i) - \prod_{i=0}^{n-1} \sin(\gamma_i).$$

Then, $\{\varphi_0, \dots, \varphi_{n-1}\} \in \mathcal{P}^n$ if and only if the function ω has a zero in the interior of the polytope P_τ for some permutation τ .

Theorem 4.1. *For $n = 3, 4, 5$, consider trees T_i , $i = 0, \dots, n - 1$ with root r , edge rv_i in T_i , $\deg(r) = 1$ in T_i , $T_i \cap T_j = \{r\}$ for $i \neq j$, such that each T_i has a greedy drawing with opening angle at least $0 < \varphi_i - \varepsilon < 180^\circ$ for any $\varepsilon > 0$. Then, tree $T = \bigcup_{i=0}^{n-1} T_i$ has a greedy drawing with $|\angle T_i| < \varphi_i$ for all $i = 0, \dots, n - 1$ if and only if $\{\varphi_0, \dots, \varphi_{n-1}\} \in \mathcal{P}^n$.*

Proof. First, consider a drawing of $K_{1,n}$ with edges rv_i that solves \mathcal{P}^n , and, without loss of generality, let the angles be ordered such that $\psi_i := \angle v_{i-1}v_i v_{i+1} < \varphi_i$. We create a greedy drawing Γ of T by drawing $(T_i)_{rv_i}^{v_i}$ arbitrarily small at v_i , such that T_i has opening angle at least $\varphi_i - \varepsilon > \psi_i$ for a sufficiently small $\varepsilon > 0$, and orienting $(T_i)_{rv_i}^{v_i}$ such that $v_j \in \angle T_i$ for all $j \neq i$; see Figure 26c.

Now assume a greedy drawing Γ_0 of T with $|\angle T_i| < \varphi_i$, $i = 0, \dots, n - 1$ exists. For one i , tree T_i might be drawn with a closed angle in Γ_0 . Then, there also exists a greedy drawing Γ , in which $0 < |\angle T_j| < \varphi_j$, $j = 0, \dots, n - 1$, by the following argument. By Lemma 4.6, the subtree $\bar{T} = \{rv_i\} + \bigcup_{j \neq i} T_j$ with root v_i must have an open angle in Γ_0 . We then obtain Γ by making the edge rv_i sufficiently long inside $\angle \bar{T}$ and drawing T_i with $|\angle T_i| > 0$, such that $\bar{T} \subseteq \angle T_i$ and $T_i \subseteq \angle \bar{T}$.

Table 3: Solving non-linear problem \mathcal{P}^n explicitly. Let $\varphi_i \geq \varphi_{i+1}$, $\varphi_i \in (0^\circ, 60^\circ) \cup (90^\circ, 120^\circ] \cup \{180^\circ\}$, $\sum_{i=0}^{n-1} \varphi_i > (n-2)180^\circ$.

n	case	$\{\varphi_0, \dots, \varphi_{n-1}\} \in \mathcal{P}^n$ iff	proof
3, 4		always	Lem. 4.19, 4.21
5	$\varphi_0 = \dots = \varphi_3 = 180^\circ$	always	Lem. 4.22
5	$\varphi_0 \leq 120^\circ$	always	Lem. 4.20
5	$\varphi_0 = \dots = \varphi_2 = 180^\circ$	$\varphi_3 + \varphi_4 > 120^\circ$	Lem. 4.23
5	$\varphi_0 = \varphi_1 = 180^\circ$	$\varphi_2 + \varphi_3 + \varphi_4 > 240^\circ$	Lem. 4.24
5	$\varphi_0 = 180^\circ$, $\varphi_1, \varphi_2, \varphi_3 \in (90^\circ, 120^\circ]$, $\varphi_4 \leq 60^\circ$?	
5	$\varphi_0 = 180^\circ$, $\varphi_1, \dots, \varphi_4 \in (90^\circ, 120^\circ]$?	

Now let all T_i have open angles in Γ_0 . Let Λ_i be the cone $\angle T_i$ in Γ_0 , and let $x_i = \text{apex}(\angle T_i)$. By Lemma 4.5, for $i \neq j$, point x_j lies in the interior of Λ_i . Therefore, angle $\angle x_{i-1}x_i x_{i+1}$ (i modulo n) is less than the angle of Λ_i . We apply Lemma 4.9 to T_0 , then to T_1, \dots, T_{n-1} and obtain a greedy drawing Γ' of $K_{1,n}$ formed by segments rx_i for $i = 0, \dots, n-1$. For $n = 4, 5$, for each pair of consecutive edges rx_i, rx_j in Γ' the turn from rx_i to rx_j is less than 180° , so r lies inside the convex polygon with corners x_0, \dots, x_{n-1} . Therefore, Γ' directly provides a solution of \mathcal{P}^n . For $n = 3$, x_1 might lie inside angle $\angle x_0rx_2 \leq 180^\circ$. However, since $\varphi_0 + \varphi_1 + \varphi_2 > 180^\circ$, we have $\{\varphi_0, \varphi_1, \varphi_2\} \in \mathcal{P}^3$; see Lemma 4.19. \square

Although Problem (*) is non-linear, we are almost always able to give tight conditions for the existence of the solution; see Table 3, which summarizes Lemmas 4.19 to 4.24.

Lemma 4.19. For angles $\varphi_0, \varphi_1, \varphi_2 > 0$, $\sum_{i=0}^2 \varphi_i > 180^\circ$, it holds: $\{\varphi_0, \varphi_1, \varphi_2\} \in \mathcal{P}^3$.

Proof. It is possible to choose $0 < \psi_i < \varphi_i$, $i = 0, \dots, 2$, such that $\sum_{i=0}^2 \psi_i = 180^\circ$. In Problem (*), we set $\beta_i = \gamma_{i+1} = \frac{\psi_i}{2}$; see Figure 27a. We have $\beta_i + \gamma_i = \frac{1}{2}(\psi_i + \psi_{i-1}) < 90^\circ < \alpha_i$. Therefore, this angle assignment satisfies the constraints in Problem (*) for some positive ε . \square

Lemma 4.20. For $n = 4, 5$ and angles $\varphi_0, \dots, \varphi_{n-1} \leq 120^\circ$, $\sum_{i=0}^{n-1} \varphi_i > 180^\circ(n-2)$, it holds: $(\varphi_0, \dots, \varphi_{n-1}) \in \mathcal{P}^n$.

Proof. It is possible to choose $\psi_i > 0$, $i = 0, \dots, n-1$ such that $\psi_i < \varphi_i$, $\sum_{i=0}^{n-1} \psi_i = 180^\circ(n-2)$. Again, we set $\beta_i = \gamma_{i+1} = \frac{\psi_i}{2}$ in Problem (*). All these angles are less than 60° , and all constraints in (*) are satisfied. Then, the corresponding drawing in Figure 26a provides a solution. \square

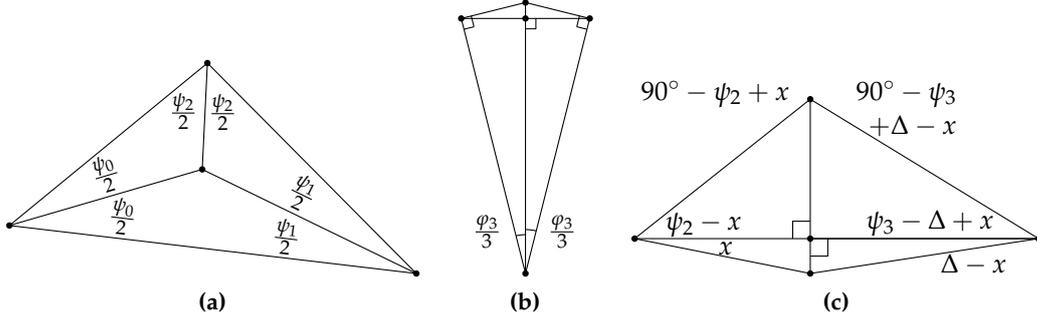


Figure 27: (a) A solution to \mathcal{P}^3 . (b), (c): solutions of \mathcal{P}^4 for $\varphi_0 = 180^\circ$.

Lemma 4.21. Consider angles $\varphi_0, \dots, \varphi_3$, such that for each $i = 0, \dots, 3$,

$$\varphi_i \in (0^\circ, 60^\circ] \cup (90^\circ, 120^\circ] \cup \{180^\circ\}.$$

If $\sum_{i=0}^3 \varphi_i > 360^\circ$, then $\{\varphi_0, \dots, \varphi_3\} \in \mathcal{P}^4$.

Proof. Without loss of generality, let $\varphi_i \geq \varphi_{i+1} > 0^\circ$. If $\varphi_0 < 180^\circ$, then $\varphi_0 \leq 120^\circ$, and the statement holds by Lemma 4.20. Let $\varphi_0 = 180^\circ$. If $\varphi_3 > 90^\circ$, then a square is a solution of \mathcal{P}^4 by choosing $\beta_i = \gamma_i = 45^\circ$. Hence, let $\varphi_3 \leq 60^\circ$. Then, $\varphi_1 + \varphi_2 > 120^\circ$, and $\varphi_1 > 90^\circ$.

If $\varphi_2 > 90^\circ$, then the construction in Figure 27b provides a solution. Let now $\varphi_2 \leq 60^\circ$. Since we have $\sum_{i=1}^3 \varphi_i > 180^\circ$, there exist $0^\circ < \psi_i < \varphi_i$, $i = 1, \dots, 3$, and $0^\circ < \Delta < \varphi_3$, such that $\sum_{i=1}^3 \psi_i = 180^\circ + \Delta$. Consider the angle assignment in Figure 27c. For $x \in (0, \Delta)$, all angles β_j, γ_j in Problem (*) are in $(0^\circ, 90^\circ)$, and all α_i are 90° . Consider the function

$$\begin{aligned} f(x) &= \sin(\psi_2 - x) \sin(90^\circ - x) \sin(\Delta - x) \sin(90^\circ - \psi_3 + \Delta - x) \\ &\quad - \sin(x) \sin(90^\circ - \Delta + x) \sin(\psi_3 - \Delta + x) \sin(90^\circ - \psi_2 + x). \end{aligned}$$

We have $f(0) > 0$ and $f(\Delta) < 0$. Hence, for some $x \in (0, \Delta)$ we have $f(x) = 0$. For this value of x , the angle assignment provides a solution of \mathcal{P}^4 . \square

Lemma 4.22. For each $\varphi_4 > 0^\circ$, $\{180^\circ, 180^\circ, 180^\circ, 180^\circ, \varphi_4\} \in \mathcal{P}^5$.

Proof. Let $0^\circ < 16\delta < \min\{\varphi_4, 60^\circ\}$. The following angle assignment solves (*):

$$\begin{aligned} \beta_0 &= \gamma_1 = 8\delta, \quad \gamma_0 = \beta_1 = 90^\circ - 5\delta, \quad \alpha_0 = \alpha_1 = 90^\circ - 3\delta, \\ \beta_i &= \gamma_i = 60^\circ - \delta, \quad \alpha_i = 60^\circ + 2\delta, \quad i = 2, \dots, 4. \end{aligned} \quad \square$$

Lemma 4.23. For $\varphi_0 = \varphi_1 = \varphi_2 = 180^\circ$, $\varphi_3, \varphi_4 \leq 120^\circ$, we have

$$\{\varphi_0, \dots, \varphi_4\} \in \mathcal{P}^5 \text{ if and only if } \varphi_3 + \varphi_4 > 120^\circ.$$

Proof. First, let $\varphi_3 + \varphi_4 \leq 120^\circ$. Consider the opening angles in the two embedding options in Figure 28a and 28b. In the first case, angles with strict upper

bounds 60° , φ_3 and φ_4 must pairwise contain apices of each other. In the second case, consider the triangle Δ formed by the lines $\text{axis}(rv_0)$, $\text{axis}(rv_1)$ and $\text{axis}(rv_2)$. By Lemma 4.14, the angle of Δ formed by $\text{axis}(rv_0)$ and $\text{axis}(rv_1)$ is less than $\frac{\varphi_3}{2}$ and the angle of Δ formed by $\text{axis}(rv_0)$ and $\text{axis}(rv_2)$ is less than $\frac{\varphi_4}{2}$. Furthermore, the angle of Δ formed by $\text{axis}(rv_1)$ and $\text{axis}(rv_2)$ is less than 120° . Thus, in both cases, the sum of the three angles is below 180° , a contradiction.

Now, let $\varphi_3 + \varphi_4 > 120^\circ$, $\varphi_3, \varphi_4 \leq 120^\circ$. There exist $\psi_i < \varphi_i$, $i = 3, 4$, and a sufficiently small $\delta > 0$, such that $\psi_3 + \psi_4 - 8\delta > 120^\circ$. Then, the following assignment satisfies (*):

$$\begin{aligned} \beta_1 = \gamma_2 &= \frac{\psi_3}{2} < 60^\circ, \quad \beta_3 = \gamma_4 = \frac{\psi_4}{2} < 60^\circ \\ \beta_2 = \gamma_1 &= 90^\circ - \frac{\psi_3}{4} - \delta, \quad \alpha_1 = \alpha_2 = 90^\circ - \frac{\psi_3}{4} + \delta \\ \beta_4 = \gamma_3 &= 90^\circ - \frac{\psi_4}{4} - \delta, \quad \alpha_3 = \alpha_4 = 90^\circ - \frac{\psi_4}{4} + \delta \\ \beta_0 = \gamma_0 &= 90^\circ - \frac{\psi_3 + \psi_4}{4} + 2\delta, \quad \alpha_0 = \frac{\psi_3 + \psi_4}{2} - 4\delta > 60^\circ. \end{aligned} \quad \square$$

Lemma 4.24. *Let $\varphi_2, \varphi_3 \in (90^\circ, 120^\circ]$ and $\varphi_4 \in (0^\circ, 120^\circ]$, $\varphi_2 \geq \varphi_3 \geq \varphi_4$. Then,*

$$\{180^\circ, 180^\circ, \varphi_2, \varphi_3, \varphi_4\} \in \mathcal{P}^5 \text{ if and only if } \varphi_2 + \varphi_3 + \varphi_4 > 240^\circ.$$

Proof. First, let $\varphi_2 + \varphi_3 + \varphi_4 \leq 240^\circ$. Then, the embedding option in Figure 29a is not possible, since $120^\circ + \varphi_2 + \varphi_3 + \varphi_4 \leq 360^\circ$. Thus, the only possible embedding option is the one in Figure 29b for $\{\varphi_2, \varphi_3, \varphi_4\} = \{\alpha, \beta, \gamma\}$. Assume a solution of \mathcal{P}^5 exists, and consider the corresponding construction in Figure 29c. The bounds $\frac{\alpha}{2}$, $\frac{x+\beta}{2}$ and $\frac{y+\gamma}{2}$ result from the fact that the two thick subsegments in Figure 29c have at most half the length of the corresponding segment. We must have

$$\frac{\alpha}{2} + \frac{x+\beta}{2} + \frac{y+\gamma}{2} > 180^\circ \Rightarrow x+y > 120^\circ \Rightarrow z = 180^\circ - x - y < 60^\circ,$$

a contradiction to $z > \max\{x, y\}$.

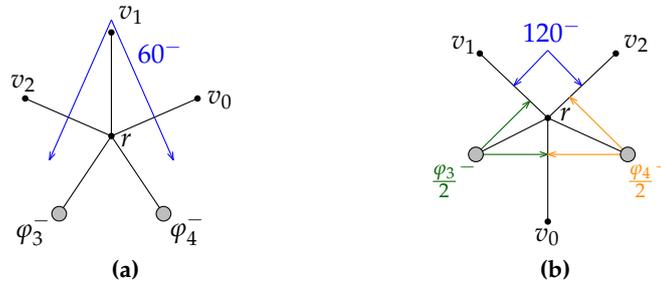


Figure 28: Proof of Lemma 4.23. Both orderings are not possible if $\varphi_3 + \varphi_4 \leq 120^\circ$.

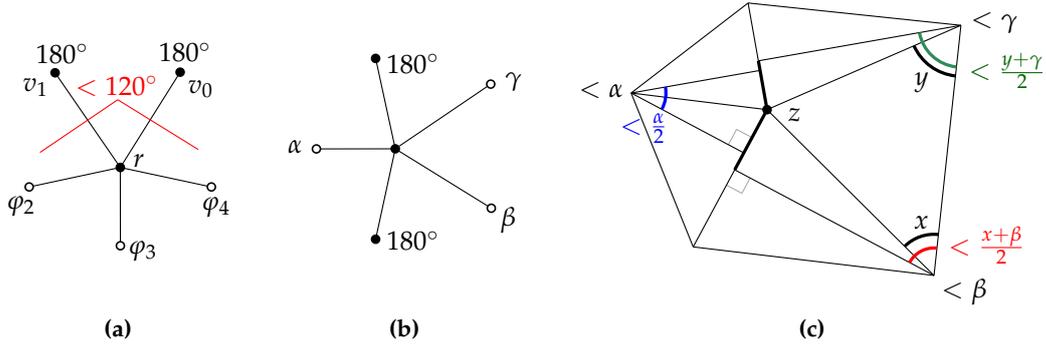


Figure 29: Proof of Lemma 4.24: $\varphi_2 + \varphi_3 + \varphi_4 > 240^\circ$ is necessary.

Now, let $\varphi_2 + \varphi_3 + \varphi_4 > 240^\circ$. Then, there exist ψ_2, ψ_3, ψ_4 and a sufficiently small δ such that:

$$\begin{aligned} \psi_i &< \varphi_i, \quad i = 2, 3, 4; \\ \psi_2 &> 90^\circ, \quad \psi_3 > 90^\circ, \quad \psi_4 < 60^\circ; \\ \delta &< \min\{\psi_4, 30^\circ\}; \\ \psi_j - 60^\circ + \delta &< 60^\circ, \quad j = 2, 3; \\ \psi_2 + \psi_3 + \psi_4 - 8\delta &= 240^\circ. \end{aligned}$$

Let such $\psi_2, \psi_3, \psi_4, \delta$ be fixed. For $x \in (0^\circ, \psi_4)$, consider the angle assignment shown in Figure 30. We have

$$(90^\circ - \frac{\psi_4}{2} + \frac{x}{2} + \delta) - (\psi_4 - x) = 90^\circ - \frac{3}{2}\psi_4 + \frac{3}{2}x + \delta > 0^\circ.$$

Further, we have

$$(90^\circ - \frac{x}{2} + \delta) - x = 90^\circ - \frac{3}{2}x + \delta > 0^\circ.$$

Thus, it can be easily verified that this angle assignment satisfies all linear constraints in (*). It remains to show the existence of $x \in (0^\circ, \psi_4)$ such that:

$$\begin{aligned} f(x) &:= \sin(x) \sin(120^\circ - \frac{\psi_2}{2} - \delta) \sin(\psi_3 - 60^\circ + \delta) \sin(90^\circ - \frac{\psi_4}{2} + \frac{x}{2} - \delta) \\ &\quad - \sin(\psi_4 - x) \sin(90^\circ - \frac{x}{2} - \delta) \sin(\psi_2 - 60^\circ + \delta) \sin(120^\circ - \frac{\psi_3}{2} - \delta) \stackrel{!}{=} 0. \end{aligned}$$

For $x = 0^\circ$, all angles in the above term, except x , are in $(0^\circ, 90^\circ)$. Similarly, for $x = \psi_4$, all angles except $\psi_4 - x$ are in $(0^\circ, 90^\circ)$. Therefore, we have $f(0) < 0$ and $f(\psi_4) > 0$. Thus, such x exists. \square

The last two cases for $n = 5$ in Table 3 are the only remaining ones to consider (for $\varphi_3 + \varphi_4 > 120^\circ$, $\varphi_2 + \dots + \varphi_4 > 240^\circ$, $\varphi_1 + \dots + \varphi_4 > 360^\circ$). In practice, it is possible to either strictly prove $\{\varphi_0, \dots, \varphi_4\} \notin \mathcal{P}^5$ or numerically construct

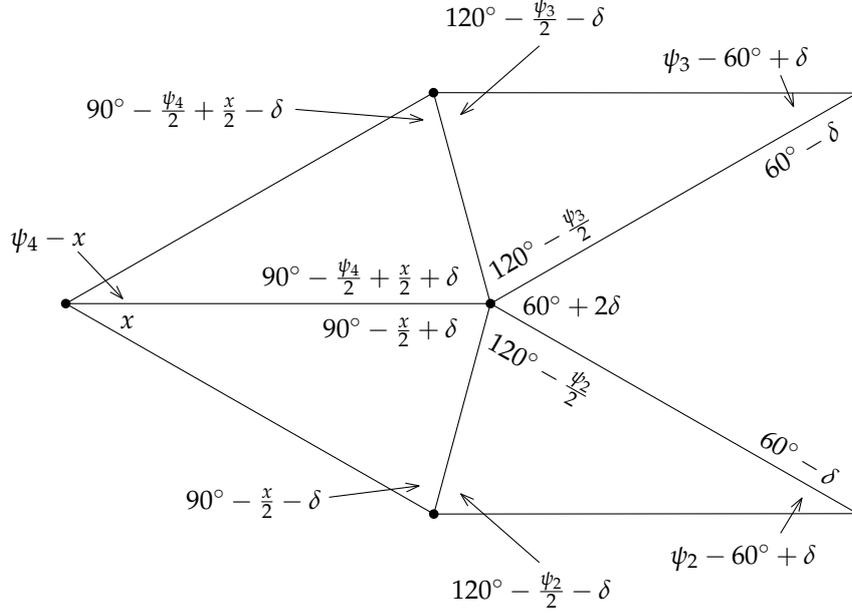


Figure 30: Proof of Lemma 4.24: $\varphi_2 + \varphi_3 + \varphi_4 > 240^\circ$ is sufficient.

a solution for many such sets of angles. If we drop the last constraint in (*), we acquire a linear program which has a constant number of variables and constraints and can be solved in $O(1)$. If it has no solution for any cyclic order of φ_i , neither has \mathcal{P}^5 . For example, this is the case for $\{180^\circ, 105^\circ, 105^\circ, 105^\circ, 60^\circ\}$; see Figure 34a. If this linear program has a solution, we can try to solve (*) using nonlinear programming solvers. For example, using MATLAB we solved \mathcal{P}^5 for the tree in Figure 34b; a solution is shown in Figure 34c. However, if the solver finds no solution, we obviously have no guarantee that none exists. For example, this was the case for $\{180^\circ, 120^\circ, 120^\circ, 120^\circ, 30^\circ\}$; see Figure 34d.

Lemma 4.25 presents a sufficient condition for the first of the two above cases. We do not know whether it is also necessary, but interestingly, in our experiments, MATLAB found a solution exactly when it was satisfied.

Lemma 4.25. Consider angles $0^\circ \leq \varphi_4 \leq 60^\circ$, $90^\circ < \varphi_3 \leq \varphi_2 \leq \varphi_1 \leq 120^\circ$, $\varphi_1 + \dots + \varphi_4 > 360^\circ$. Let the following two conditions hold:

- (i) $14\varphi_1 + 12\varphi_2 + 8\varphi_3 + 15\varphi_4 > 4500^\circ$
- (ii) For $x := \min\{\frac{1}{7}(14\varphi_1 + 12\varphi_2 + 8\varphi_3 + 15\varphi_4 - 4500^\circ), \varphi_4\}$ and $p_1 \in [0^\circ, 90^\circ]^{10}$, $p_1 = (\beta_0, \dots, \beta_4, \gamma_0, \dots, \gamma_4)$ defined as:

$$\begin{aligned} \beta_0 &= \varphi_4 - x, \\ \beta_1 &= 90^\circ - \frac{x}{2}, \\ \beta_2 &= \varphi_3 + \frac{\varphi_2}{2} + \frac{\varphi_1}{4} + \frac{\varphi_4 - x}{8} - 157.5^\circ, \\ \beta_3 &= \varphi_2 + \frac{\varphi_1}{2} + \frac{\varphi_4 - x}{4} - 135^\circ, \end{aligned}$$

$$\begin{aligned} \beta_4 &= \varphi_1 - 90^\circ + \frac{\varphi_4 - x}{2}, \\ \gamma_0 &= 90^\circ - \frac{\varphi_4 - x}{2}, \\ \gamma_1 &= x, \\ \gamma_2 &= 168.75^\circ - \frac{\varphi_3}{2} - \frac{\varphi_2}{4} - \frac{\varphi_1}{8} - \frac{\varphi_4 - x}{16}, \\ \gamma_3 &= 157.5^\circ - \frac{\varphi_2}{2} - \frac{\varphi_1}{4} - \frac{\varphi_4 - x}{8}, \\ \gamma_4 &= 135^\circ - \frac{\varphi_1}{2} - \frac{\varphi_4 - x}{4}, \end{aligned}$$

it holds: $\omega(p_1) < 0$.

Then, $\{180^\circ, \varphi_1, \dots, \varphi_4\} \in \mathcal{P}^5$.

The proof can be found in Appendix A.

4.5 CHARACTERIZING GREEDY-DRAWABLE BINARY TREES

In this section, we shall characterize greedy-drawable binary trees by forbidden subgraphs.

Let us consider the following subtree Q_k with root b_0 . It consists of nodes $b_0, b_1, c_1, b_2, c_2, \dots, b_{k+1}, c_{k+1}, b_{k+2}$. Node b_i is connected to b_{i-1}, b_{i+1} and c_i , and nodes c_1, \dots, c_k and b_{k+1} are leaves (e.g., $Q_0 = K_{1,3}$). Figure 31a shows a subdivision of such a subtree Q_1 .

Lemma 4.26. For a subdivision of Q_k , an open angle $\psi_k \geq 90^\circ + 30^\circ / (2^k)$ is not possible. For each $\varepsilon' > 0$, Q_k can be drawn with open angle $90^\circ + 30^\circ / (2^k) - \varepsilon'$.

Proof. We have $|\angle Q_0| < 120^\circ$. Each subdivision of Q_0 contains it as a subgraph. Furthermore, angle $120^\circ - \varepsilon$ is possible for every $\varepsilon > 0$ (draw each of the three simple paths collinear and make all edges arbitrarily small, except for the three segments adjacent to the node of degree 3).

Let T_k be a subdivision of Q_k , $k \geq 1$. Let $(b_0 = x_0, x_1, \dots, x_p, b_1)$ be a subdivision of b_0b_1 , $(b_1, y_1, \dots, y_q, b_2)$ be a subdivision of b_1b_2 and $(b_1, z_1, \dots, z_r, c_1)$ a subdivi-

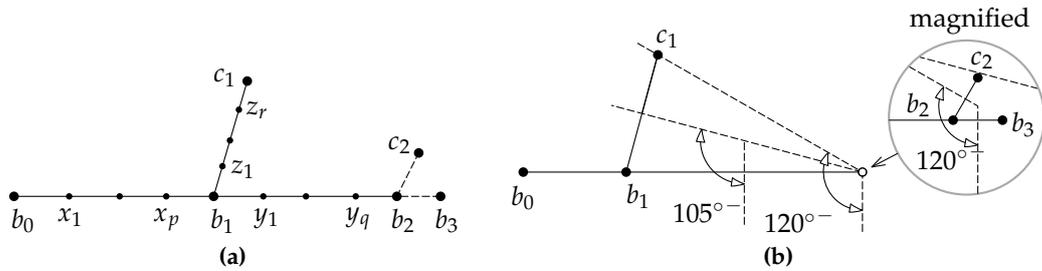


Figure 31: A subdivision of Q_1 and a greedy drawing with nearly optimal opening angle.

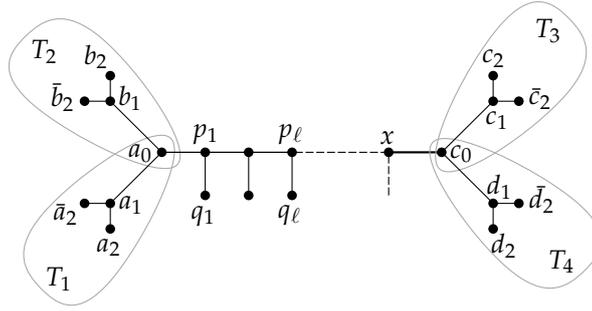


Figure 32: Sketch of the proof of Lemma 4.27.

sion of c_1c_2 ; see Figure 31a. Let $T_{k-1} = (T_k)_{b_1y_1}^{y_1} + b_1y_1$ with root b_1 . Then, T_{k-1} is a subdivision of Q_{k-1} , and by induction, $|\angle T_{k-1}| = (90^\circ - \frac{30^\circ}{2^{k-1}})^-$. Applying Lemma 4.13 to $(T_k)_{x_p b_1}^{b_1} + x_p b_1$ and then repeatedly applying Lemma 4.12 to $(T_k)_{x_i x_{i+1}}^{x_{i+1}} + x_i x_{i+1}$ provides

$$|\angle T_k| = (\frac{1}{2}(90^\circ - \frac{30^\circ}{2^{k-1}}) + 45^\circ)^- = (90^\circ - \frac{30^\circ}{2^k})^-. \quad \square$$

For $\varepsilon > 0$, angle $90^\circ + 30^\circ / (2^k) - \varepsilon$ is achieved if b_1, b_2, \dots, b_{k+2} lie on a single line, $\angle b_{k+2}b_{k+1}c_{k+1}$ is slightly bigger than 60° , $\angle b_{k+1}b_k c_k$ is slightly bigger than 75° etc; see Figure 31b.

It follows that subtrees of type Q_k and subdivisions thereof can always be drawn with an opening angle $90 + \varepsilon_k$, $\varepsilon_k > 0$, for any fixed k . We show that if there are at most four such independent components in T , their four open angles can always be arranged appropriately.

Lemma 4.27. *If a binary tree T contains at most four independent stars of degree 3 each having a leaf as a root (i.e., $n_3 \leq 4$; see Section 4.2.1), it has a greedy drawing.*

Proof. Without loss of generality, let $n_3 = 4$. Then, T contains a subdivision of a subtree depicted in Figure 32. (Coincidentally, for $\ell = 0$, this is exactly the “crab” from [Ala+13]. Hence, T has no self-approaching drawing.) Let $T_1 = T_{a_0 a_1}^{a_1} + a_0 a_1$, $T_2 = T_{a_0 b_1}^{b_1} + a_0 b_1$ be subtrees of T rooted at a_0 and $T_3 = T_{c_0 c_1}^{c_1} + c_0 c_1$, $T_4 = T_{c_0 d_1}^{d_1} + c_0 d_1$ be subtrees of T rooted at c_0 . Then, T_1, \dots, T_4 must be subdivisions of caterpillars of type Q_k (otherwise it would be $n_3 \geq 5$).

Now, we start combining the subtrees. Let $T' = T_{c_0 x}^x + x c_0$ and $T'' = T_{x c_0}^{c_0} + x c_0$. By applying Lemmas 4.12 and 4.13 to T_1 and T_2 as well as to T_3 and T_4 , it follows that both T' and T'' can be drawn greedily with an opening angle ε for sufficiently small $\varepsilon > 0$. We apply Lemma 4.11 to T' and T'' and then merge the two drawings at edge $x c_0$ and gain a greedy drawing of T . \square

Recall that for $n_3 \geq 6$, no greedy drawing exists; see Section 4.2.1. We now consider the remaining case $n_3 = 5$. In this case, T must contain a *five-crab* subgraph shown in Figure 33a or a subdivision thereof. We consider the corresponding independent subtrees T_1, \dots, T_5 of T . Again, these subtrees must be

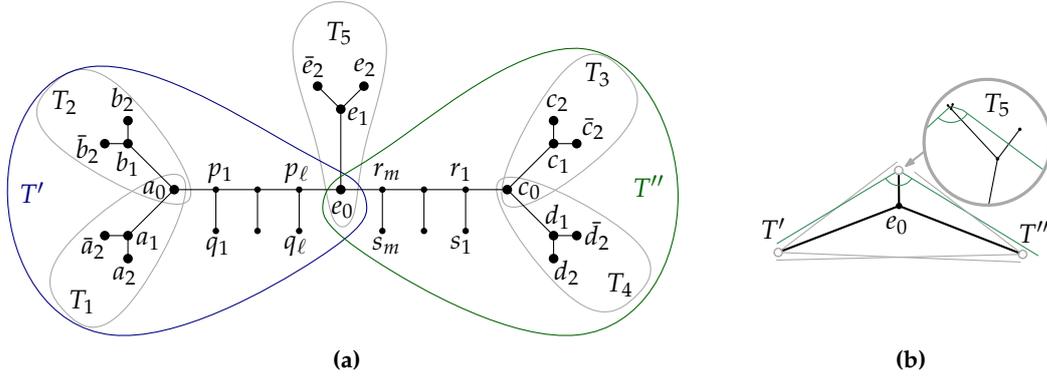


Figure 33: Constructing a greedy drawing for $n_5 = 5$.

caterpillars of type Q_k , otherwise, $n_3 \geq 6$. Each T_i can be drawn with an opening angle $|\angle T_i| = \varphi_i^-$ for $\varphi_i \in (90^\circ, 120^\circ]$. Let $\sigma = \sum_{i=1}^5 \varphi_i$. If $\sigma \leq 540^\circ$, no greedy drawing exists by Lemma 4.7.

We now show that a greedy drawing always exists for $\sigma > 540^\circ$. Similar to the proof of Lemma 4.27, we combine the subtrees T_1, T_2 , the a_0 - e_0 -path and edges $p_1q_1, \dots, p_\ell q_\ell$ to the subtree $T' = T_{e_0p_\ell}^{p_\ell} + e_0p_\ell$ as well as the subtrees T_3, T_4 , the c_0 - e_0 -path and edges r_1s_1, \dots, r_ms_m to the subtree $T'' = T_{e_0r_m}^{r_m} + e_0r_m$ (both with root e_0). By applying Lemmas 4.12, 4.13 and 4.15, we have $|\angle T'| = \varphi'^-$ for $\varphi' = \varphi_1 + \varphi_2 - 180^\circ$ and $|\angle T''| = \varphi''^-$ for $\varphi'' = \varphi_3 + \varphi_4 - 180^\circ$. Since $\varphi' + \varphi'' + \varphi_5 > 180^\circ$, we have $\{\varphi', \varphi'', \varphi_5\} \in \mathcal{P}^3$ by Lemma 4.19.

We now list all the possibilities for $\sigma > 540^\circ$. For the rooted subtrees $T_i, i = 1, \dots, 5$ we say that T_i has order k if T_i is equivalent to a subdivision of Q_k . Assume at least four of the five subtrees have order 1 or greater, then $\sigma \leq 120^\circ + 4 \cdot 105^\circ = 540^\circ$, so T cannot be drawn greedily. Thus, at least two subtrees T_i have order 0. If there are three, four or five such subtrees of order 0, then $\sigma > 3 \cdot 120^\circ + 2 \cdot 90^\circ = 540^\circ$. If there are only two, then at least two of the three remaining subtrees have order 1, if a greedy drawing exists, since $2 \cdot 120^\circ + 105^\circ + 2 \cdot 97.5^\circ = 540^\circ$. In this case, $\sigma > 2 \cdot 120^\circ + 2 \cdot 105^\circ + 90^\circ = 540^\circ$. In both cases, i.e., (i) at least three subtrees of order 0 or (ii) two subtrees of order 0 and at least two subtrees of order 1, a greedy drawing exists by Lemma 4.20.

We can now give a complete characterization of greedy-drawable trees with maximum degree 3.

Proposition 4.1. *A tree T with maximum degree 3 has a greedy drawing in \mathbb{R}^2 if and only if one of the following holds:*

- (i) $n_3 \leq 4$, or
- (ii) T contains a subdivision of a five-crab in Figure 33a, such that the rooted subtrees T_1, \dots, T_5 as defined above are subdivisions of Q_k with the orders either $\{0, 0, 0, x_1, x_2\}$ or $\{0, 0, 1, 1, x_1\}$ for some $x_1, x_2 \in \mathbb{N}_0$.

Alternatively, we can express it using forbidden subgraphs: (i) a five-crab with four subtrees Q_1 or (ii) a five-crab with two subtrees Q_2 and one Q_1 (or subdivisions thereof).

4.6 RECOGNITION ALGORITHM

4.6.1 Maximum degree 4

In this section we formulate Algorithm 1, which decides for a tree T with maximum degree 4 whether T has a greedy drawing. First, we describe a procedure to determine the tight upper bound for the opening angle of a given rooted subtree. After processing a node v , we set a flag $p(v) = \text{true}$. Let $N_p(v) = \{u \mid uv \in E, p(u) = \text{true}\}$, and $\angle_{\text{optimal}}(\varphi_1, \varphi_2, \varphi_3)$ the new tight upper bound calculated according to Table 2. For the ease of description, when we consider a single edge or a path, we say that the tight upper bound on its opening angle is 180° .

Lemma 4.28. *Procedure `getOpenAngle` is correct and requires time $O(|V|)$.*

Proof. The algorithm processes tree nodes bottom-up. For $v \in V$, let π_v be the parent of v , $\deg(v) = d_v$, $T_v = T_{\pi_v}^v + \pi_v v$ with root π_v . For a subtree with one node, define its opening angle as 180° . We prove the following invariant for the *while* loop. For each $v \in V$ with $p(v) = \text{true}$, $\angle(v) > 0$ stores a tight upper bound for the opening angle in a greedy drawing of T_v .

The invariant holds for all leaves of T after the initialization. The first *if*-statement inside the *while* body ensures that if all nodes in T_v except v have degree 1 or 2, then $\angle(v) = 180$ if $d_v = 1, 2$ in T , $\angle(v) = 120$ if $d_v = 3$ and $\angle(v) = 60$ if $d_v = 4$. Now consider the first *else* clause inside the *while* loop. Assume $p(v) = \text{false}$, $|N_p(v)| = d_v - 1$ and the invariant holds for all subtrees T_u , $u \in N_p(v)$. If one of the cases I–V can be applied to v and subtrees T_u , then, after the current loop, $\angle(v) > 0$ stores the tight upper bound for the opening angle in a greedy drawing of T_v ; see Table 2. Otherwise, we have case VI or VII, and T_v cannot be drawn with an open angle. Each node v is processed in $O(d_v)$, and if for $u \in N(v) - N_p(v)$, it holds $|N_p(u)| \geq d_u - 1$ after processing v , we put u in a queue. Hence the running time is $O(|V|)$. \square

Proposition 4.2. *Algorithm 1 is correct and requires time $O(|V|)$.*

Proof. The algorithm is similar to Procedure `getOpenAngle`, except that T now does not have a distinguished root. We proceed from the leaves of T inwards. For a node v with $|N_p(v)| = d_v - 1$, let $\{r_v\} = N(v) - N_p(v)$. Similar to Procedure `getOpenAngle`, after $p(v)$ is set true, $\angle(v) > 0$ stores the tight upper bound for the opening angle of subtree $T_{r_v}^v + r_v v$ (this is proved as in Lemma 4.28).

Let us now consider the two *return* statements. In the first one, we have a node v , and for all its neighbors u_i , $i = 0, \dots, d_v - 1$, $p(u_i) = \text{true}$ and $\varphi_i = \angle(u_i) > 0$ by the invariant. Angle φ_i is the tight upper bound on the opening angle for

Procedure getOpenAngle(T, r)**Input** : tree $T = (V, E)$, root $r \in V$, $d_r = 1$ **Output**: tight upper bound on $|\angle T|$, 0 if no open angle possible. $p(r) \leftarrow \text{false}$ **for** $v \in V \setminus \{r\}$ **do** **if** $d_v = 5$ **then return** 0 **else if** $d_v = 1$ **then** $p(v) \leftarrow \text{true}; \angle(v) \leftarrow 180$ **else** $p(v) \leftarrow \text{false}$ **while** $\exists v \in V : \neg p(v) \ \& \ |N_p(v)| = d_v - 1$ **do** **if** $\forall u \in N_p(v) : \angle(u) = 180$ **then** $\angle(v) \leftarrow 180 - (d_v - 2) \cdot 60$ **else if case I, ..., V applicable then** $\angle(v) \leftarrow \angle_{\text{optimal}}(\angle(u_1), \dots, \angle(u_k))$, for $\{u_1, \dots, u_k\} = N_p(v)$ **else return** 0 $p(v) \leftarrow \text{true}$ **return** $\angle(v)$ for $\{v\} = N(r)$ **Algorithm 1:** hasGreedyDrawing(T)**Input** : tree $T = (V, E)$ with maximum degree 4**Output**: whether T has a greedy drawing**for** $v \in V$ **do** **if** $d_v = 1$ **then** $p(v) \leftarrow \text{true}; \angle(v) \leftarrow 180$ **else** $p(v) \leftarrow \text{false}$ **while** $\exists v \in V : \neg p(v) \ \& \ |N_p(v)| \geq d_v - 1$ **do** **if** $|N_p(v)| = d_v$ **then** **return** $\sum_{u, uv \in E} \angle(u) > (d_v - 2)180$ **else if** $\forall u \in N_p(v) : \angle(u) = 180$ **then** $\angle(v) \leftarrow 180 - (d_v - 2) \cdot 60$ **else if case I, ..., V applicable then** $\angle(v) \leftarrow \angle_{\text{optimal}}(\angle(u_1), \dots, \angle(u_k))$, for $\{u_1, \dots, u_k\} = N_p(v)$ **else** $w \leftarrow N(v) - N_p(v)$ $\angle(w) \leftarrow \text{getOpenAngle}(T_{vw}^w + vw, v)$ **return** $\angle(w) > 0 \ \& \ \sum_{u, uv \in E} \angle(u) > (d_v - 2)180$ $p(v) \leftarrow \text{true}$

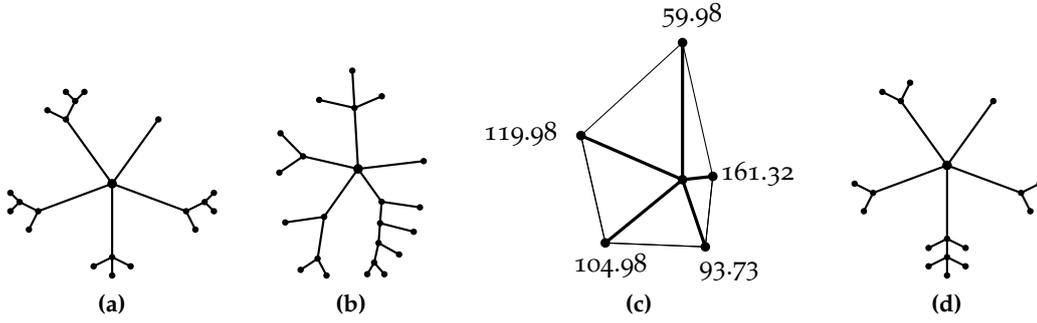


Figure 34: Examples of trees with a node of degree 5. Tree (a) has no greedy drawing, since $\{180^\circ, 105^\circ, 105^\circ, 105^\circ, 60^\circ\} \notin \mathcal{P}^5$. Tree (b) has one, since $\{180^\circ, 120^\circ, 105^\circ, 93.75^\circ, 60^\circ\} \in \mathcal{P}^5$, see the solution in (c) found by a non-linear solver. It is not clear whether the tree in (d) has a greedy drawing. By Theorem 4.1, proving existence is equivalent to deciding whether $\{180^\circ, 120^\circ, 120^\circ, 120^\circ, 30^\circ\} \in \mathcal{P}^5$.

the subtree $T_{vu_i}^{u_i} + u_i v$. Hence, if $\sigma = \sum_{i=0}^{d_v-1} \varphi_i \leq (d_v - 2)180^\circ$, by Lemma 4.7, no greedy drawing of T exists. Now let $\sigma > (d_v - 2)180^\circ$. If $d_v = 2$, the two opening angles can be arranged in a suitable way. If $d_v = 3, 4$, then $\{\varphi_0, \dots, \varphi_{d_v-1}\} \in \mathcal{P}^{d_v}$; see Table 3. By Theorem 4.1, a greedy drawing of T exists.

Finally, consider the second *return* statement and the last *else* clause of the algorithm. Let $\{u_0, \dots, u_{d-2}\} = N_p(v)$ and $\varphi_i = \angle(u_i)$. Again, since none of the cases I–V is applicable, the combined tree $T_{vw}^v + vw$ with root w must have a closed angle. Hence, if $\angle(w) = 0$, $T_{vw}^w + vw$ must also form a closed angle, and no greedy drawing exists by Lemma 4.6. Now let $\varphi_{d-1} = \angle(w) > 0$, $\sigma = \sum_{i=0}^{d-1} \varphi_i$. Similar to the previous case, a greedy drawing exists if and only if $\sigma > (d - 2)180^\circ$; see Table 3. \square

4.6.2 Maximum degree 5 and above

If T contains a node v with $\deg(v) \geq 6$, no greedy drawing exists. Also, a greedy-drawable tree can have at most one node of degree 5 by Lemma 4.8, otherwise, there are two independent stars each having 5 leaves.

For unique $r \in V$, $\deg(r) = 5$, consider the five rooted subtrees T_0, \dots, T_4 attached to it and the tight upper bounds φ_i on $|\angle T_i|$. If $\sigma = \sum_{i=0}^4 \varphi_i \leq 540^\circ$, T cannot be drawn greedily. The converse, however, does not hold. By Theorem 4.1, a greedy drawing exists if and only if $\{\varphi_0, \dots, \varphi_4\} \in \mathcal{P}^5$. To decide whether $\{\varphi_0, \dots, \varphi_4\} \in \mathcal{P}^5$, we apply the conditions from Table 3. For the remaining case $\varphi_0 = 180^\circ$, $\varphi_1, \dots, \varphi_4 \leq 120^\circ$, if the sufficient condition of Lemma 4.25 does not apply and the linear relaxation of Problem (*) has a solution, but the non-linear solver finds none for Problem (*), we report *uncertain*; see Algorithm 2. An uncertain example is presented in Figure 34d.

Algorithm 2: Deciding if a tree with maximum degree 5 has a greedy drawing.

```

Input : tree  $T = (V, E)$  with maximum degree 5,  $r \in V, \deg(r) = 5$ .
Output: whether  $T$  has a greedy drawing
if  $\exists u \in V \setminus \{r\}, \deg(u) = 5$  then
    return false
 $(u_0, \dots, u_4) \leftarrow N(r)$ 
for  $i = 0, \dots, 4$  do
     $\alpha_i \leftarrow \text{getOpenAngle}(T_{ru_i}^{u_i} + ru_i, r)$ 
    if  $\alpha_i = 0$  then return false
if  $\sum_{i=0}^4 \alpha_i \leq 540$  then return false
 $(\varphi_0, \dots, \varphi_4) \leftarrow \text{SortDescending}(\alpha_0, \dots, \alpha_4)$ 
if  $\varphi_0 \leq 120$  then return true
if  $\varphi_3 = 180$  then return true
if  $\varphi_2 = 180$  then return  $\varphi_3 + \varphi_4 > 120$ 
if  $\varphi_1 = 180$  then return  $\varphi_2 + \varphi_3 + \varphi_4 > 240$ 
if  $\varphi_4 \leq 60$  & condition in Lemma 4.25 holds then return true
if no LP has a solution then return false
if solved  $\{\varphi_0, \dots, \varphi_4\} \in \mathcal{P}^5$  numerically then return true
// cases for which we have no guarantee for  $\{\varphi_0, \dots, \varphi_4\} \notin \mathcal{P}^5$ 
return uncertain

```

4.7 CONCLUSION

In this chapter, we gave the first complete characterization of all trees that admit a greedy embedding in \mathbb{R}^2 with the Euclidean distance metric, thereby solving the corresponding open problem stated by Angelini et al. [ADF12]. This is a further step in characterizing the graphs that have Euclidean greedy embeddings.

Open questions

One direction of future work is to develop heuristics to actually draw greedy trees with non-zero edge lengths. Some simple strategies can be derived from the proofs presented in this chapter. However, optimizing the resolution of such drawings appears to be a challenging task.

To fill the gaps in the characterization of graphs with an Euclidean greedy embedding in \mathbb{R}^2 , it would be interesting to consider other graph classes, e.g., non-3-connected planar graphs with cycles. Another challenging question is to describe graphs with planar or convex greedy drawings. For example, the still-open strong Papadimitriou-Ratajczak conjecture [PR05] states that every 3-connected planar graph has a planar greedy drawing with convex faces.

ON SELF-APPROACHING AND INCREASING-CHORD DRAWINGS OF 3-CONNECTED PLANAR GRAPHS

Recall that an st -path in a drawing of a graph is *self-approaching* if during the traversal of the corresponding curve from s to any point t' on the curve the distance to t' is non-increasing. A path is *increasing-chord* if it is self-approaching in both directions. A drawing is self-approaching (increasing-chord) if any pair of vertices is connected by a self-approaching (increasing-chord) path.

In this chapter, we study self-approaching and increasing-chord drawings of triangulations and 3-connected planar graphs. We show that in the Euclidean plane, triangulations admit increasing-chord drawings, and for planar 3-trees we can ensure planarity. We prove that strongly monotone (and thus increasing-chord) drawings of trees and binary cactuses require exponential resolution in the worst case, answering an open question by Kindermann et al. [Kin+14]. Moreover, we provide a binary cactus that does not admit a self-approaching drawing. Additionally, we show that 3-connected planar graphs admit increasing-chord drawings in the hyperbolic plane and characterize the trees that admit such drawings.

This chapter is based on joint work with Martin Nöllenburg and Ignaz Rutter [NPR14; NPR16].

5.1 INTRODUCTION

A popular use case of graph drawings is to support users in the task of finding paths in a network. In this setting, the notions of geodesic-path tendency and path continuity described in Chapter 2 have been empirically shown to be important design criteria for graph drawings. A number of different graph drawing styles implementing these notions have been introduced, namely greedy drawings [Rao+03] considered in Chapter 4, (strongly) monotone drawings [Ang+12] as well as *self-approaching* and *increasing-chord drawings* [Ala+13]. For an overview of the related works concerned with these drawing styles, we again refer to Chapter 2.

5.1.1 *Contribution*

We obtain the following results on constructing self-approaching or increasing-chord drawings.

1. We show that every triangulation has an increasing-chord drawing (answering an open question of Alamdari et al. [Ala+13]) and construct a *binary cactus* that does not admit a self-approaching drawing (Section 5.3). The latter is a notable difference to greedy drawings, since every binary cactus has a greedy drawing. This has been proved by Leighton and Moitra [LM10] and Angelini et al. [AFG10] as the essential result for constructing greedy drawings of 3-connected planar graphs. We also prove that strongly monotone (and, thus, increasing-chord) drawings of trees and binary cactuses require exponential resolution in the worst case, answering an open question by Kindermann et al. [Kin+14]. Using the developed techniques, we show an analogous result for greedy drawings of binary cactuses, which proves a conjecture by Leighton and Moitra [MLo8, slide 79].
2. We show how to construct plane increasing-chord drawings for *planar 3-trees* (a special class of triangulations) using Schnyder realizers (Section 5.4). Very recently, Felsner et al. [Fel+16] showed how to construct planar strongly monotone drawings for all 3-connected planar graphs, and Da Lozzo et al. [DDF17] showed the corresponding result for planar greedy drawings. The existence of planar self-approaching drawings, however, remains open even for triangulations.
3. We show that, similar to the greedy case [Kle07], the hyperbolic plane \mathbb{H}^2 allows representing a broader class of graphs than \mathbb{R}^2 (Section 5.5). We prove that a tree has a self-approaching or increasing-chord drawing in \mathbb{H}^2 if and only if it either has maximum degree 3 or is a subdivision of $K_{1,4}$ (this is not the case in \mathbb{R}^2 ; see the characterization by Alamdari et al. [Ala+13]), implying that every 3-connected planar graph has an increasing-chord drawing. (Barnette proved [Bar66] that 3-connected planar graphs can always be spanned by binary trees.) We also show how to construct planar increasing-chord drawings of binary cactuses in \mathbb{H}^2 .
4. Finally, we use *generalized* self-approaching curves [Aic+01] to prove that Euclidean greedy drawings of trees and cactuses have bounded dilation.

5.2 PRELIMINARIES

half-plane \mathbf{h}_p^q

For points $p, q \in \mathbb{R}^2$, $p \neq q$, let \mathbf{h}_p^q denote the half-plane not containing p bounded by the line through q orthogonal to the segment pq . (Note that this is different from h_{pq}^q used in Chapter 4.) A piecewise-smooth curve is self-approaching if and only if for each point a on the curve, the line perpendicular to the curve at a does not intersect the curve at a later point [IKL99]. This leads to the following characterization of self-approaching paths.

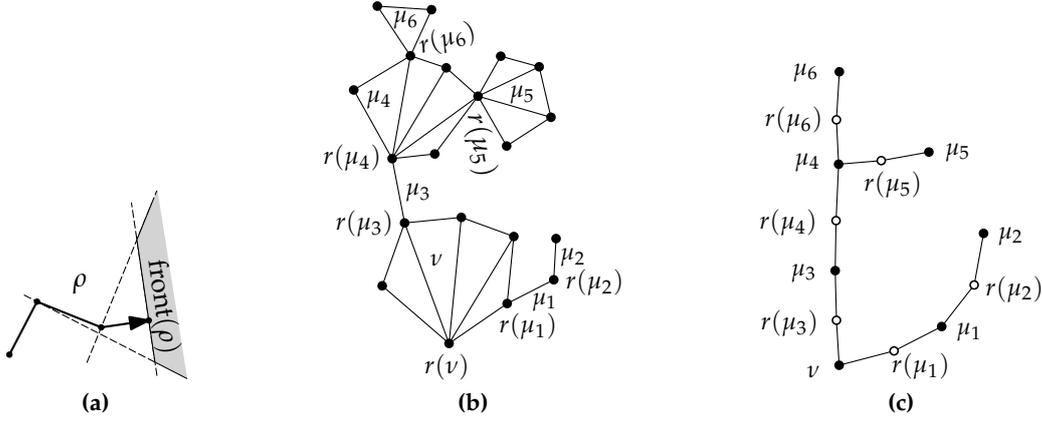


Figure 35: (a) self-approaching path ρ and $\text{front}(\rho)$ (gray area). (b), (c): downward-triangulated binary cactus and the corresponding BC-tree. B-nodes are black, C-nodes white.

Fact 5.1 (Corollary 2 in [Ala+13]). *Let $\rho = (v_1, v_2, \dots, v_k)$ be a directed path embedded in \mathbb{R}^2 with straight-line segments. Then, ρ is self-approaching if and only if for all $1 \leq i < j \leq k$, the point v_j lies in $\mathbf{h}_{v_i}^{v_{i+1}}$.*

We shall denote the reverse of a path ρ by ρ^{-1} . Let $\rho = (v_1, v_2, \dots, v_k)$ be a self-approaching path. Define $\text{front}(\rho) = \bigcap_{i=1}^{k-1} \mathbf{h}_{v_i}^{v_{i+1}}$, see also Figure 35a. Using Fact 5.1, we can decide whether a concatenation of two paths is self-approaching.

$\text{front}(\rho)$

Fact 5.2 (Concatenating self-approaching paths). *Let $\rho_1 = (v_1, \dots, v_k)$ and $\rho_2 = (v_k, v_{k+1}, \dots, v_m)$ be self-approaching paths. The path $\rho_1 \cdot \rho_2 := (v_1, \dots, v_k, v_{k+1}, \dots, v_m)$ is self-approaching if and only if $\rho_2 \subseteq \text{front}(\rho_1)$.*

The following result concerning increasing-chord paths can be obtained as a corollary of Lemma 3 in [IKL99].

Lemma 5.1. *Let $\rho = (v_1, \dots, v_k)$ be a path such that for any $i < j$, $i, j \in \{1, \dots, k-1\}$, we have $\angle(\overrightarrow{v_i v_{i+1}}, \overrightarrow{v_j v_{j+1}}) \leq 90^\circ$. Then, ρ is increasing-chord.*

Proof. For any $j > i$, $i, j \in \{1, \dots, k-1\}$, we have $\angle(\overrightarrow{v_{j+1} v_j}, \overrightarrow{v_{i+1} v_i}) \leq 90^\circ$. Thus, the condition of the lemma also holds for ρ^{-1} , and by symmetry it is sufficient to prove that ρ is self-approaching.

We claim that for each $i \in \{1, \dots, k-1\}$ and each $j \in \{i+1, \dots, k\}$, we have $v_j \in \mathbf{h}_{v_i}^{v_{i+1}}$. Once the claim is proved, it follows from Fact 5.1 that ρ is self-approaching. For the proof of the claim let $i \in \{1, \dots, k-1\}$ be arbitrary and fixed. It suffices to show that $v_{i+2}, \dots, v_k \in \mathbf{h}_{v_i}^{v_{i+1}}$.

First consider v_{i+2} . By the condition of the lemma, $\angle(\overrightarrow{v_i v_{i+1}}, \overrightarrow{v_{i+1} v_{i+2}}) \leq 90^\circ$. Therefore, $v_{i+2} \in \mathbf{h}_{v_i}^{v_{i+1}}$. Now assume $v_j \in \mathbf{h}_{v_i}^{v_{i+1}}$ for some $j \in \{i+2, \dots, k-1\}$. We show $v_{j+1} \in \mathbf{h}_{v_i}^{v_{i+1}}$. Consider the half-plane $h \subseteq \mathbf{h}_{v_i}^{v_{i+1}}$ whose boundary is parallel to that of $\mathbf{h}_{v_i}^{v_{i+1}}$ and contains v_j . Since $\angle(\overrightarrow{v_i v_{i+1}}, \overrightarrow{v_j v_{j+1}}) \leq 90^\circ$, we have $v_{j+1} \in h \subseteq \mathbf{h}_{v_i}^{v_{i+1}}$. \square

block Let $G = (V, E)$ be a connected graph. Recall that a *block* is a maximal biconnected subgraph. The *block-cutvertex tree* (or *BC-tree*) T_G of G has a *B-node* for each block of G , a *C-node* for each cutvertex of G and, for each block v containing a cutvertex v , an edge between the corresponding *B-* and *C-node*. We associate *B-nodes* with their corresponding blocks and *C-nodes* with their corresponding cutvertices.

parent block $\pi(\mu)$ The following notation follows the work of Angelini et al. [AFG10]. Let T_G be rooted at some block v containing a non-cutvertex (such a block v always exists). For each block $\mu \neq v$, let $\pi(\mu)$ denote the *parent block* of μ , i.e., the grandparent of μ in T_G . Let $\pi^2(\mu)$ denote the parent block of $\pi(\mu)$ and, generally, $\pi^{i+1}(\mu)$ the parent block of $\pi^i(\mu)$. Further, we define the *root* $r(\mu)$ of μ as the cutvertex contained in both μ and $\pi(\mu)$. Note that $r(\mu)$ is the parent of μ in T_G . In addition, for the root node v of T_G , we define $r(v)$ to be some non-cutvertex of v . Let $\text{depth}_B(\mu)$ denote the number of *B-nodes* on the v - μ -path in T_G minus 1, and let $\text{depth}_C(r(\mu)) = \text{depth}_B(\mu)$. If μ is a leaf of T_G , we call it a *leaf block*.

leaf block Recall that a *cactus* is a graph in which every edge is part of at most one simple cycle. Note that every cactus is outerplanar. In a *binary* (or *Christmas*) cactus every cutvertex is part of exactly two blocks. For a binary cactus G with a block μ containing a cutvertex v , let G_μ^v denote the maximal connected subgraph containing v but no other vertex of μ . We say that G_μ^v is a *subcactus* of G . Let G be a binary cactus with a fixed root and let v be a cutvertex of G . Then the block μ containing v such that $v \neq r(\mu)$ is unique, and we write G^v for G_μ^v .

triangular fan A *triangulated cactus* is a cactus together with additional edges, which make each of the cactus blocks internally triangulated. A *triangular fan with vertices* $V_t = \{v_0, v_1, \dots, v_k\}$ and root v_0 is a graph on V_t with edges $v_i v_{i+1}$, $i = 1, \dots, k-1$, as well as $v_0 v_i$, $i = 1, \dots, k$. Let us consider a special kind of triangulated cactuses, each of whose blocks μ is a triangular fan with root $r(\mu)$. We call such a cactus *downward-triangulated* and every edge of a block μ incident to $r(\mu)$ a *downward edge*. Figure 35b and 35c show a downward-triangulated binary cactus and the corresponding BC-tree.

downward edge Consider a fixed straight-line drawing of a cactus G with root r . We define the set of *upward directed edges*

$$E_U(G) = \{r(\mu)v \mid \mu \text{ is a block of } G \text{ containing } v, v \neq r(\mu)\}.$$

Note that if G is not triangulated, some edges in $E_U(G)$ might not be edges in G . If G is binary, then, for cutvertex u , let U_u denote the upward directed edges of the subcactus rooted at u or, formally, $U_u = E_U(G^u)$.

upward and downward directions $U(G), D(G)$

Additionally, we define the set of *upward directions*

$$U(G) = \{\overrightarrow{r(\mu)v} \mid \mu \text{ is a block of } G \text{ containing } v, v \neq r(\mu)\}$$

and the set of *downward directions*

$$D(G) = \{\overrightarrow{uv} \mid \overrightarrow{vu} \in U(G)\}.$$

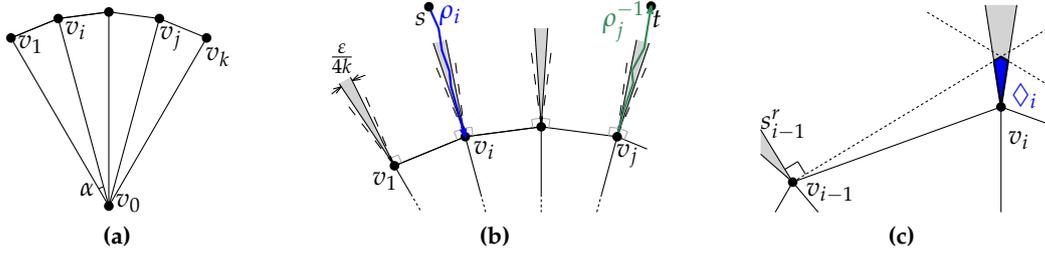


Figure 36: Drawing a triangulated binary cactus with increasing chords inductively. The drawings $\Gamma_{i,\epsilon'}$ of the subcactuses, $\epsilon' = \frac{\epsilon}{4k}$, are contained inside the gray cones. We have $\beta = 90^\circ - \epsilon'$, $\gamma = 90^\circ + \epsilon'/2$.

5.3 GRAPHS WITH SELF-APPROACHING DRAWINGS

A natural approach to construct (not necessarily plane) self-approaching drawings is to construct a self-approaching drawing of a spanning subgraph. For instance, to draw a graph G containing a Hamiltonian path H with increasing chords, we simply draw H consecutively on a line. In this section, we consider 3-connected planar graphs and the special case of triangulations, which addresses an open question of Alamdari et al. [Ala+13]. These graphs are known to have a spanning binary cactus [AFG10; LM10]. Angelini et al. [AFG10] showed that every triangulation has a spanning downward-triangulated binary cactus.

5.3.1 Increasing-chord drawings of triangulations

We show that all downward-triangulated binary cactuses have increasing-chord drawings. The construction is similar to the one of the greedy drawings of binary cactuses in the two proofs of the Papadimitriou-Ratajczak conjecture [LM10; AFG10]. Our proof is by induction on the height of the BC-tree. We show that G can be drawn such that all downward edges are almost vertical and the remaining edges almost horizontal. Then, for vertices s, t of G , an st -path with increasing chords goes downwards to some block μ , then sideways to another cutvertex of μ and, finally, upwards to t . Let \vec{e}_1, \vec{e}_2 be vectors $(1, 0)^\top, (0, 1)^\top$ respectively.

Theorem 5.1. *Let $G = (V, E)$ be a downward-triangulated binary cactus. For any $0^\circ < \epsilon < 90^\circ$, there exists an increasing-chord drawing Γ_ϵ of G , such that for each vertex v contained in some block μ , $v \neq r(\mu)$, the angle formed by $\overrightarrow{r(\mu)v}$ and \vec{e}_2 is less than $\frac{\epsilon}{2}$.*

Proof. Let G be rooted at block v . As our base case, let $v = G$ be a triangular fan with vertices v_0, v_1, \dots, v_k and root $v_0 = r(v)$. We draw v_0 at the origin and distribute v_1, \dots, v_k on the unit circle, such that $\angle(\vec{e}_2, \overrightarrow{v_0v_1}) = k\alpha/2$ and $\angle(\overrightarrow{v_0v_i}, \overrightarrow{v_0v_{i+1}}) = \alpha$, $\alpha = \epsilon/2k$; see Figure 36a. By Lemma 5.1, path (v_1, \dots, v_k) has increasing chords.

Now let G have multiple blocks. We draw the root block v , $v_0 = r(v)$, as in the previous case, but with $\alpha = \frac{\epsilon}{2k}$. Then, for each $i = 1, \dots, k$, we choose $\epsilon' = \frac{\epsilon}{4k}$ and

draw the subcactus $G_i = G_v^{v_i}$ rooted at v_i inductively, such that the corresponding drawing $\Gamma_{i,\varepsilon'}$ is aligned at $\overrightarrow{v_0 v_i}$ instead of \vec{e}_2 ; see Figure 36b. Note that ε' is the angle of the cones (gray) containing $\Gamma_{i,\varepsilon'}$. Obviously, all downward edges of G form angles less than $\frac{\varepsilon}{2}$ with \vec{e}_2 .

We must be able to reach any t in any G_j from any s in any G_i via an increasing-chord path ρ . To achieve this, we make sure that no normal on a downward edge of G_i crosses the drawing of G_j , $j \neq i$. Let Λ_i be the cone with apex v_i and angle ε' aligned with $\overrightarrow{v_0 v_i}$, $v_0 \notin \Lambda_i$ (gray regions in Figure 36b). Let s_i^l and s_i^r be the left and right boundary rays of Λ_i with respect to the direction of $\overrightarrow{v_0 v_i}$, and h_i^l, h_i^r the half-planes with boundaries containing v_i and orthogonal to s_i^l and s_i^r respectively, such that $v_0 \in h_i^l \cap h_i^r$. For $i = 2, \dots, k-1$, define $\diamond_i = \Lambda_i \cap h_{i-1}^r \cap h_{i+1}^l$ (thin blue quadrilateral in Figure 36c). Let $\diamond_1 = \Lambda_1 \cap h_2^l$ and $\diamond_k = \Lambda_k \cap h_{k-1}^r$. For any $i, j = 1, \dots, k$, $i \neq j$, we have $\diamond_j \subseteq h_i^r \cap h_i^l$. We now scale each drawing $\Gamma_{i,\varepsilon'}$ such that it is contained in \diamond_i . In particular, for any downward edge uv in $\Gamma_{i,\varepsilon'}$, we have $\Gamma_{j,\varepsilon'} \subseteq \diamond_j \subseteq \mathbf{h}_u^v$ for $j \neq i$. We claim that the resulting drawing of G is an increasing-chord drawing.

Consider vertices s, t of G . If s and t are contained in the same subgraph G_i , an increasing-chord st -path in G_i exists by induction. If s is in G_i and t is v_0 , let ρ_i be the s - v_i -path in G_i that uses only downward edges. By Lemma 5.1, path ρ_i is increasing-chord and remains so after adding edge $v_i v_0$.

Finally, assume t is in G_j with $j \neq i$. Let ρ_j be the t - v_j -path in G_j that uses only downward edges. Due to the choice of ε' , $h_i^r \cap h_i^l \subseteq \text{front}(\rho_i)$ contains v_1, \dots, v_k in its interior. Consider the path $\rho' = (v_i, v_{i+1}, \dots, v_j)$. It is self-approaching by Lemma 5.1; also, $\rho' \subseteq \text{front}(\rho_i)$ and $\rho_j \subseteq \text{front}(\rho')$. We also have $\rho_j \subseteq \diamond_j \subseteq \text{front}(\rho_i)$. Fact 5.2 lets us concatenate ρ_i, ρ' and ρ_j^{-1} to a self-approaching path. By a symmetric argument, it is also self-approaching in the opposite direction and, thus, is increasing-chord. \square

Since every triangulation has a spanning downward-triangulated binary cactus [AFG10], this implies that planar triangulations admit increasing-chord drawings.

Corollary 5.1. *Every planar triangulation admits an increasing-chord drawing.*

5.3.2 Exponential worst case resolution

The construction for a spanning downward-triangulated binary cactus in Section 5.3.1 requires exponential area. In this section, we show that we cannot do better in the worst case even for strongly monotone drawings of downward-triangulated binary cactuses. Recall that increasing-chord drawings are strongly monotone.

For the following lemma, we want to point out the difference between a *greedy st-path* and a *greedy drawing* of a graph G , such that G is a path. In a fixed drawing, an st -path $\rho = (v_0 = s, v_1, \dots, v_k, v_{k+1} = t)$ is greedy (or *distance-decreasing*), if $|v_{i+1}t| < |v_i t|$ for every $i = 0, \dots, k$. Note that for some $0 \leq i < j \leq k+1$,

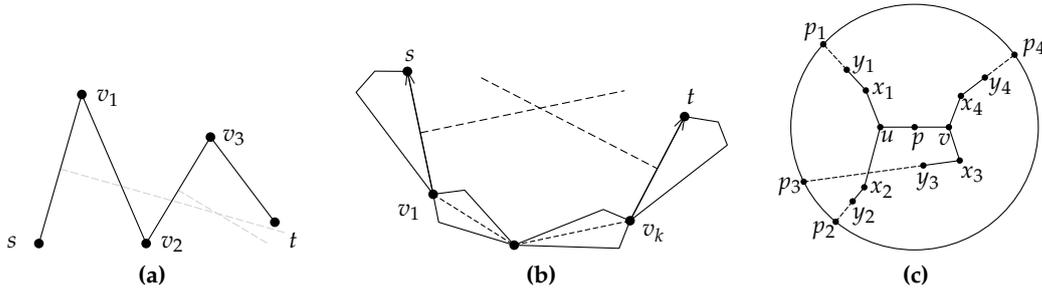


Figure 37: (a) The st -path (s, v_1, v_2, v_3, t) is a greedy path, but its s - v_2 -subpath is not. Thus, this drawing is not a greedy drawing of a path. (b) Proof of Lemma 5.2. (c) Proof of Lemma 5.3.

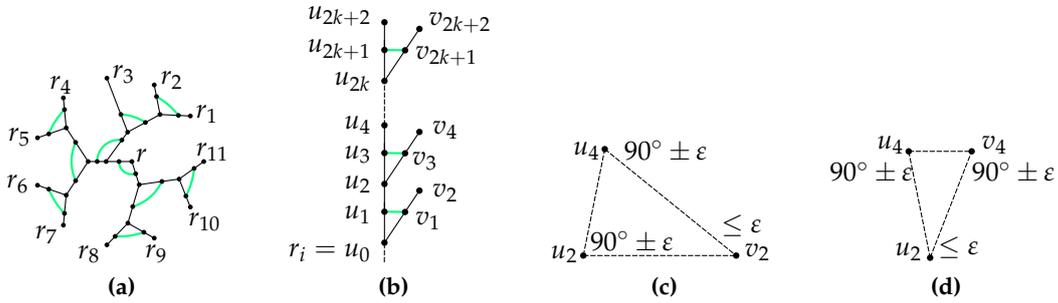


Figure 38: Family of binary cactuses G_k requiring exponential area for any strongly monotone drawing. (a) Central cactus G' ; (b) binary subcactus C_k attached to each vertex of degree 1 of G' . In a strongly monotone drawing of G_k , we have: (c) $|u_2u_4| \leq |u_2v_2| \tan \varepsilon$; (d) $|u_4v_4| \leq |u_2u_4| \tan \varepsilon$.

$\{i, j\} \neq \{0, k + 1\}$, the v_i - v_j -path $(v_i, v_{i+1}, \dots, v_j)$ is not necessarily greedy; see Figure 37a. On the other hand, for a graph G which is a path $\rho = (v_0, v_1, \dots, v_k, v_{k+1})$, a drawing Γ is a greedy drawing of G if every v_i - v_j -path $(v_i, v_{i+1}, \dots, v_j)$ and every v_j - v_i -path $(v_j, v_{j-1}, \dots, v_i)$ in Γ is a greedy path for any $0 \leq i < j \leq k + 1$.

The following lemma describes directions of certain edges in a greedy or monotone drawing of a cactus.

Lemma 5.2. For a cactus $G = (V, E)$ and two vertices $s, t \in V$, consider the cutvertices v_1, \dots, v_k lying on every simple st -path in G in this order. In any greedy drawing of G , connecting consecutive vertices in (s, v_1, \dots, v_k, t) would form a greedy drawing of the path (s, v_1, \dots, v_k, t) . In any monotone drawing, connecting consecutive vertices in (s, v_1, \dots, v_k, t) would form a monotone drawing of the path (s, v_1, \dots, v_k, t) . In both cases, $\text{ray}(v_1, s)$ and $\text{ray}(v_k, t)$ diverge.

Proof. Let $v_0 = s, v_{k+1} = t$. For $0 \leq i < j \leq k + 1$, any v_i - v_j -path and any v_j - v_i -path in G contains vertices v_i, v_{i+1}, \dots, v_j . Since a path in a greedy drawing of G remains greedy after replacing subpaths by shortcuts, the segments $sv_1, v_1v_2, \dots, v_{k-1}v_k, v_kv_t$ form a greedy drawing. By Lemma 14 (Lemma 7 of Angelini et al. [ADF12]), $\text{ray}(v_1, s)$ and $\text{ray}(v_k, t)$ diverge; see Figure 37b.

Analogously, a path remains monotone after replacing subpaths by shortcuts. Therefore, in a monotone drawing of G , segments $sv_1, v_1v_2, \dots, v_{k-1}v_k, v_kv_t$ form a monotone drawing. Since a monotone path cannot make a turn of 180° or more, $\text{ray}(v_1, s)$ and $\text{ray}(v_k, t)$ must diverge. \square

For the following lemma, consider a greedy or monotone drawing of a binary cactus G with root r . Recall that for a cutvertex u , the set U_u denotes the upward directed edges of the subcactus rooted at u . Then, the following property holds.

Lemma 5.3. *In a monotone or greedy drawing of a binary cactus with root r , consider cutvertices $u, v \neq r$, such that the subcactuses G^u and G^v are disjoint. Then the edges in U_u and in U_v each form a single interval in the circular order induced by their joint set of directions.*

Proof. Consider four pairs of vertices $x_i, y_i, i = 1, \dots, 4$, such that $x_iy_i \in U_u$ for $i = 1, 2$ and $x_iy_i \in U_v$ for $i = 3, 4$. Note that, by the definition of U_u and U_v , vertices x_i and x_j are cutvertices. For $i = 1, 2, j = 3, 4$, let ρ_{ij} denote the vertex sequence y_i, x_i, u, v, x_j, y_j . Since x_i, u, v, x_j are cutvertices, ρ_{ij} is a subsequence of every y_i - y_j -path. By Lemma 5.2, every such ρ_{ij} forms a monotone or a greedy drawing of a path, respectively. Hence, ρ_{ij} is non-crossing and cannot make a turn of 180° or more. Additionally, by Lemma 5.2, rays $\text{ray}(x_i, y_i)$ and $\text{ray}(x_j, y_j)$ must diverge. Finally, $\text{ray}(x_j, y_j)$ cannot cross ρ_{ij} , since this would imply that ρ_{ij} has made a turn of 180° or more, and neither can $\text{ray}(x_i, y_i)$.

We define $p = (u + v)/2$ and choose an arbitrary $R > 0$, such that all paths ρ_{ij} are contained inside a circle C with center p and radius R . Let p_i be the intersection of $\text{ray}(x_i, y_i)$ and C . Assume p_1, p_3, p_2, p_4 is the counterclockwise order of p_i on the boundary of C ; see Figure 37c. Then, for some pair $i, j, i \in \{1, 2\}, j \in \{3, 4\}$, there exists a crossing of $\text{ray}(x_i, y_i)$ or $\text{ray}(x_j, y_j)$ with ρ_{ij} or with each other; a contradiction. Therefore, p_1, p_2 as well as p_3, p_4 appear consecutively on the boundary of C . Now let the radius R approach infinity. Then, $\overrightarrow{pp_i}$ becomes parallel with $\overrightarrow{x_iy_i}$. Therefore, the circular order $\overrightarrow{x_1y_1}, \overrightarrow{x_3y_3}, \overrightarrow{x_2y_2}, \overrightarrow{x_4y_4}$ is not possible, and the statement follows. \square

Note that for trees, Angelini et al. [Ang+12] call this property *slope disjointness*. Consider the following family of binary cactuses G_k . Let G' be a rooted binary cactus with eleven vertices r_1, \dots, r_{11} of degree 1 and its root r as the only vertex of degree 2; see Figure 38a. Next, consider cactus C_k consisting of a chain of $k + 1$ triangles and some additional degree-1 nodes as in Figure 38b. We construct G_k by attaching a copy of C_k to each r_i in G' . From now on, consider a strongly monotone drawing of G_k .

Using Lemma 5.3 and the pigeonhole principle, we can show the following fact.

Lemma 5.4. *For some $r_i, i \in \{1, \dots, 11\}$, each pair of directions in U_{r_i} forms an angle at most $\varepsilon = 360^\circ/11$.*

Proof. Consider the two cutvertices of the root block of G_k ; see Figure 38a. By Lemma 5.3, vectors in $U_{r_1} \cup \dots \cup U_{r_{11}}$ appear in the following circular order: first

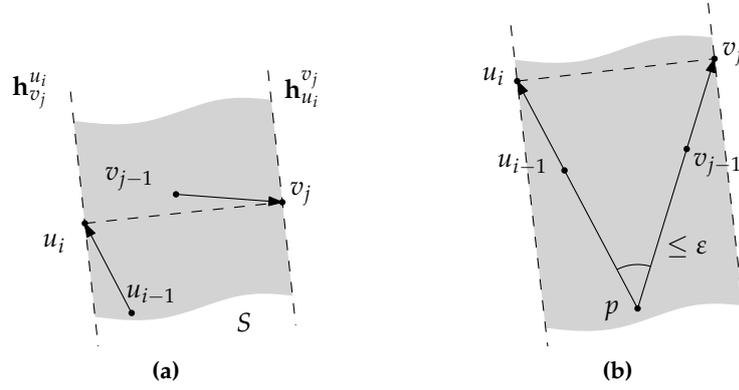


Figure 39: Proof of Lemma 5.5. (a) Points u_{i-1} and v_{j-1} must lie below $u_i v_j$ inside the strip S . (b) From the triangle $u_i v_j p$, it follows that $\angle p v_j u_i = \angle v_{j-1} v_j u_i < 90^\circ$.

the vectors in $U_{r_1} \cup \dots \cup U_{r_7}$, then the vectors in $U_{r_8} \cup \dots \cup U_{r_{11}}$. By applying the same argument to the child blocks repetitively, it follows that the vectors have the following circular order: first the vectors in $U_{r_{\pi(1)}}$, then the vectors in $U_{r_{\pi(2)'}}$, \dots , then the vectors in $U_{r_{\pi(11)}}$ for some permutation π . Therefore, for some i , each pair of directions in U_{r_i} forms an angle at most $\epsilon = 360^\circ / 11$. \square

Now consider a vertex r_i with the property of Lemma 5.4. Let the vertices of its subcactus be named as in Figure 38b. Without loss of generality, we may assume that each vector in U_{r_i} forms an angle at most $\epsilon/2$ with the upward direction \vec{e}_2 . We show that certain directions have to be almost horizontal.

Lemma 5.5. *For even $i, j, 2 \leq j \leq i \leq 2k + 2$, consider vertices u_i, v_j . Vector $\overrightarrow{u_i v_j}$ forms an angle at most $\epsilon/2$ with the horizontal axis.*

Proof. Consider a strongly monotone u_i - v_j -path ρ . Vertices $u_i, u_{i-1}, v_{j-1}, v_j$ must appear on ρ in this order. We have $\angle(\overrightarrow{u_{i-1} u_i}, \overrightarrow{v_{j-1} v_j}) \leq \epsilon$. Furthermore, by the strong monotonicity of ρ , we have $\angle u_{i-1} u_i v_j, \angle u_{i-1} v_j u_i < 90^\circ$, as well as $\angle v_{j-1} u_i v_j, \angle v_{j-1} v_j u_i < 90^\circ$.

Consider the strip $S = \mathbb{R}^2 \setminus (\mathbf{h}_{u_i}^{v_j} \cup \mathbf{h}_{v_j}^{u_i})$; see Figure 39a. From the above observation on the angles, it follows $u_{i-1}, v_{j-1} \in S$. Line segment $u_i v_j$ divides S into two parts. Assume u_{i-1} and v_{j-1} are in different parts. But then, the angle $\angle(\overrightarrow{u_{i-1} u_i}, \overrightarrow{v_{j-1} v_j})$ is at least 90° , a contradiction. Thus, u_{i-1} and v_{j-1} are in the same part, and, since $\overrightarrow{u_{i-1} u_i}, \overrightarrow{v_{j-1} v_j}$ point upwards, vertices u_{i-1} and v_{j-1} are below the line through the segment $u_i v_j$.

Let p be the intersection of the lines through $u_{i-1} u_i$ and $v_{j-1} v_j$; see Figure 39b. Point p also lies below the line through $u_i v_j$. Consider the triangle with corners u_i, v_j and p . We have $\angle u_i p v_j = \angle(\overrightarrow{u_{i-1} u_i}, \overrightarrow{v_{j-1} v_j}) \leq \epsilon$. Furthermore, we have $\angle p u_i v_j = \angle u_{i-1} u_i v_j < 90^\circ$, and $\angle p v_j u_i = \angle v_{j-1} v_j u_i < 90^\circ$. Therefore, $\angle u_{i-1} u_i v_j, \angle v_{j-1} v_j u_i \in (90^\circ - \epsilon, 90^\circ)$.

Assume $\angle(\overrightarrow{u_i v_j}, \vec{e}_1) > \epsilon/2$, and let $\overrightarrow{u_i v_j}$ point upwards and to the right. The other cases are analogous. Then, since $\angle u_{i-1} u_i v_j \in (90^\circ - \epsilon, 90^\circ)$, edge $u_{i-1} u_i$

must point upwards and to the left, and we must have $\angle(\overrightarrow{u_{i-1}u_i}, \vec{e}_2) > \varepsilon/2$, a contradiction to the above assumption on the directions of the upward edges. Therefore, the statement follows. \square

The following lemma essentially shows that G_k requires exponential resolution.

Lemma 5.6. *For $i = 2, 4, \dots, 2k$, we have $|u_{i+2}v_{i+2}| \leq (\tan \varepsilon)^2 |u_i v_i|$.*

Proof. For brevity, let $i = 2$. First, we show that $|u_2 v_2|$ is significantly larger than $|u_2 u_4|$; see Figure 38c. By Lemma 5.5, $\angle u_2 u_4 v_2 \in (90^\circ - \varepsilon, 90^\circ + \varepsilon)$. Therefore, $\sin \angle u_2 v_2 u_4 \geq \sin(90^\circ - \varepsilon)$. We have:

$$\frac{|u_2 u_4|}{|u_2 v_2|} = \frac{\sin \angle u_2 v_2 u_4}{\sin \angle u_2 u_4 v_2} \leq \frac{\sin \varepsilon}{\sin(90^\circ - \varepsilon)} = \tan \varepsilon.$$

Next, we show that $|u_2 u_4|$ is significantly larger than $|u_4 v_4|$; see Figure 38d. By Lemma 5.5, $\angle u_2 v_4 u_4 \in (90^\circ - \varepsilon, 90^\circ + \varepsilon)$. Therefore, $\sin \angle u_2 v_4 u_4 \geq \sin(90^\circ - \varepsilon)$. We have:

$$\frac{|u_4 v_4|}{|u_2 u_4|} = \frac{\sin \angle u_4 u_2 v_4}{\sin \angle u_2 v_4 u_4} \leq \frac{\sin \varepsilon}{\sin(90^\circ - \varepsilon)} = \tan \varepsilon.$$

Thus, $|u_4 v_4| \leq |u_2 u_4| \tan \varepsilon \leq |u_2 v_2| (\tan \varepsilon)^2$. \square

As a consequence of Lemma 5.6 we get $|u_{2k+2} v_{2k+2}| \leq |u_2 v_2| (\tan \varepsilon)^{2k}$. We have $(\tan \varepsilon)^2 < 0.414$. Since cactus G_k contains $n = \Theta(1) + 44k$ vertices, the following exponential lower bound holds for the resolution of strongly monotone drawings.

Theorem 5.2. *There exists an infinite family of binary cactuses with n vertices that require resolution $\Omega(2^{\frac{n}{44}})$ for any strongly monotone drawing.*

Using this result, we can construct a family of trees requiring exponential area for any strongly monotone drawing. Consider the binary spanning tree T_k of G_k created by removing the thick green edges in Figure 38a and 38b. Obviously, by Theorem 5.2 it requires resolution $\Omega(2^{\frac{n}{44}})$ for any strongly monotone drawing. This answers an open question by Kindermann et al. [Kin+14]. Replacing degree-2 vertices by shortcuts and applying a more careful analysis lets us prove the following result.

Theorem 5.3. *There exists an infinite family of binary trees with n vertices that require resolution $\Omega(2^{\frac{n}{22}})$ for any strongly monotone drawing.*

Observe that exponential worst-case resolution of strongly monotone drawings of binary cactuses is a stronger result than the corresponding statement for trees. A strongly monotone drawing of a binary cactus does not necessarily induce a strongly monotone drawing of any of its spanning trees.

Using the techniques developed in this section, we can prove a similar result for greedy drawings of binary cactuses.

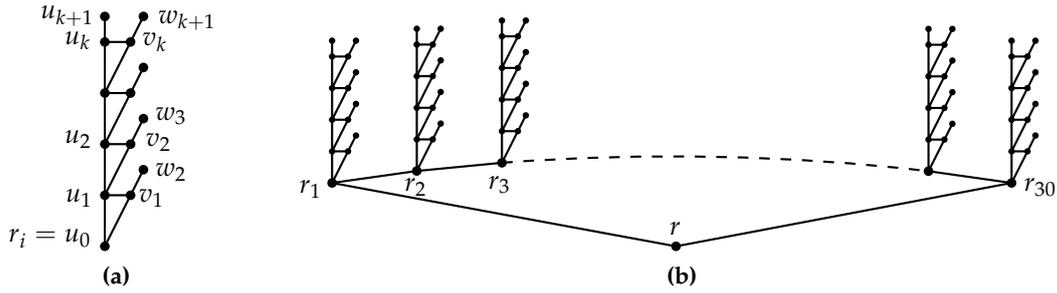


Figure 40: Family of binary cactuses that requires exponential area for every greedy embedding. (a) Cactus G_k for $k = 4$; (b) cactus F_k constructed by attaching the roots of 30 copies of G_k to a cycle of size 31.

Theorem 5.4. *There exists an infinite family of binary cactuses with n vertices that require resolution $\Omega(2^{\frac{n}{30}})$ for any greedy drawing.*

Theorem 5.4 proves a conjecture by Ankur Moitra from his presentation at the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS’08) [ML08, slide 79].

It has been proved by Leighton and Moitra [LM10] as well as by Angelini et al. [AFG10] that binary cactus graphs have Euclidean greedy drawings. This fact has played a crucial role in proving that every 3-connected planar graph has an Euclidean greedy drawing. The aforementioned proofs construct greedy drawings of binary cactuses of exponential size, and it has been an open question whether exponential area is necessary in the worst case. Theorem 5.4 shows that this is indeed the case. We recall that Goodrich and Strash [GS09] have shown how to construct an Euclidean greedy drawing of a binary cactus, in which the coordinates of every vertex can be encoded using only $O(\log n)$ bits. In that encoding scheme, vertex positions in the Euclidean plane are not stored explicitly, and the drawings might still have exponential size.

We now present a family of binary cactuses that requires exponential aspect ratio of edge lengths in every greedy embedding. For an integer $k \geq 1$, consider the binary cactus G_k with root r_i in Figure 40a. We then construct the cactus F_k by attaching the roots of 30 copies of G_k to a cycle of length 31; see Figure 40b. Let r be the root of the cactus F_k .

We shall prove that in every greedy embedding of F_k , the aspect ratio of edge lengths is at least 2^k in one of the 30 copies of G_k . The following fact can be proved by applying Lemma 5.3 to all pairs of subcactuses of F_k rooted at r_i and r_j for $i \neq j, i, j = 1, \dots, 30$.

Fact 5.3. *Every greedy embedding of F_k contains a greedy embedding of G_k , in which every pair of vectors from $\bigcup_{i=0}^k \{\overrightarrow{u_i u_{i+1}}, \overrightarrow{u_i v_{i+1}}\} \cup \bigcup_{i=1}^k \{\overrightarrow{v_i w_{i+1}}\}$ (for the naming of vertices as in Figure 40a) forms an angle of at most 12° .*

From now on, we consider an embedding of G_k satisfying the property in Fact 5.3.

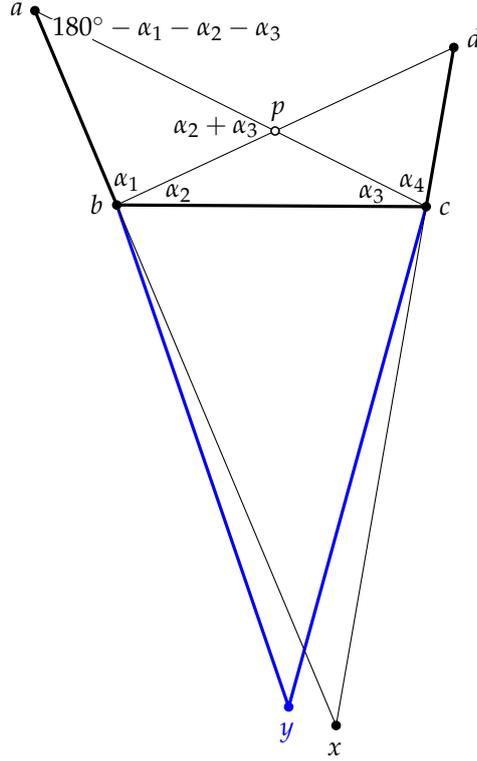


Figure 41: Proof of Lemma 5.7.

Lemma 5.7. For $0 \leq i \leq k-1$, we have $|u_{i+1}u_{i+2}| < \frac{1}{2}|u_i u_{i+1}|$.

Proof. We rename the vertices for brevity: $a = u_{i+2}$, $b = u_{i+1}$, $c = v_{i+1}$, $d = w_{i+2}$, $y = u_i$; see Figure 41. Note that since b and c are cutvertices, every distance-decreasing a - d -path as well as every distance-decreasing d - a -path must contain b and c . Therefore, the path (a, b, c, d) is a greedy drawing. Thus, the rays $\text{ray}(b, a)$ and $\text{ray}(c, d)$ diverge by Lemma 4.3. The paths (a, b, d) and (a, c, d) are also distance-decreasing in both directions, therefore, $\alpha_1 = \angle abd > 60^\circ$ and $\alpha_4 = \angle acd > 60^\circ$.

Let x be the intersection point of the lines through ab and cd . Let $\varepsilon = 12^\circ$. Since G_k has been chosen according to Fact 5.3, $\angle xby \leq \varepsilon$ and $\angle xcy \leq \varepsilon$.

We have $\angle cbx = 180^\circ - \angle abc < 120^\circ$. Similarly, $\angle bcx < 120^\circ$. Also, $\angle bxc \leq \varepsilon$. Thus, by considering the triangle bcx it follows: $\angle cbx > 60^\circ - \varepsilon$ and $\angle bcx > 60^\circ - \varepsilon$. Since $60^\circ - \varepsilon < \angle cbx < 120^\circ$, we have $60^\circ - 2\varepsilon < \angle cby < 120^\circ + \varepsilon$. Analogously, $60^\circ - 2\varepsilon < \angle bcy < 120^\circ + \varepsilon$. It follows:

$$\frac{|bc|}{|by|} = \frac{\sin \angle byc}{\sin \angle bcy} < \frac{\sin \varepsilon}{\sin(60^\circ - 2\varepsilon)} < 0.36.$$

Therefore, we have $|bc| < 0.36|by|$.

Next, recall that we have $\angle bxc = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - 180^\circ \leq \varepsilon$, for $\alpha_2 = \angle dbc$ and $\alpha_3 = \angle acb$. Therefore, $\angle bac = 180^\circ - \alpha_1 - \alpha_2 - \alpha_3 \geq \alpha_4 - \varepsilon > 60^\circ - \varepsilon$. Also,

since the path (a, b, c) is distance-decreasing in both directions, we have $\angle bac < 90^\circ$. Now consider $\angle acb = \alpha_3$. Since $\angle bcx > 60^\circ - \varepsilon$, we have $\alpha_3 + \alpha_4 < 120^\circ + \varepsilon$, and $\alpha_3 < 60^\circ + \varepsilon$. Therefore,

$$\frac{|ab|}{|bc|} = \frac{\sin \angle acb}{\sin \angle bac} = \frac{\sin \alpha_3}{\sin(180^\circ - \alpha_1 - \alpha_2 - \alpha_3)} < \frac{\sin(60^\circ + \varepsilon)}{\sin(60^\circ - \varepsilon)} < 1.28.$$

Thus, $|ab| < 1.28|bc|$. It follows: $|ab| < 1.28|bc| < 1.28 \cdot 0.36|by| < 0.461|by|$. Therefore, we have $|u_{i+1}u_{i+2}| < \frac{1}{2}|u_iu_{i+1}|$. \square

The following proposition directly implies Theorem 5.4.

Proposition 5.1. *In every greedy embedding of cactus F_k , the ratio of the longest and the shortest edge is in $\Omega(2^{n/90})$, where n is the number of vertices of F_k .*

Proof. Cactus G_k has $3k + 2$ vertices. Thus, cactus F_k has $n = 90k + 61$ vertices. By Lemma 5.7, every greedy embedding of F_k contains an embedding of G_k , such that $|u_ku_{k+1}| < \frac{1}{2^k}|u_0u_1|$. Therefore, the ratio of the longest and shortest edge in every greedy embedding of F_k is at least $2^k = \Omega(2^{n/90})$. \square

5.3.3 Non-triangulated cactuses

The construction for an increasing-chord drawing from Section 5.3.1 fails if the blocks are not triangular fans since we now cannot just use downward edges to reach the common ancestor block. Consider the family of rooted binary cactuses $G_n = (V_n, E_n)$ defined as follows. Graph G_0 is a single 4-cycle, where an arbitrary vertex is designated as the root. For $n \geq 1$, consider two disjoint copies of G_{n-1} with roots a_0 and c_0 . We create G_n by adding new vertices r_0 and b_0 both adjacent to a_0 and c_0 ; see Figure 42a. For the new block ν containing r_0, a_0, b_0, c_0 , we set $r(\nu) = r_0$. We select r_0 as the root of G_n and ν as its root block. For a block μ_i with root r_i , let a_i, b_i, c_i be its remaining vertices, such that $b_i r_i \notin E_n$. For a given drawing, due to the symmetry of G_n , we can rename the vertices a_i and c_i such that $\angle_{\text{ccw}}(\vec{r_i c_i}, \vec{r_i a_i}) \leq 180^\circ$. If a fixed block μ_i is considered, we refer to a_i, b_i, c_i as a, b, c for brevity. We now prove the following negative result.

Theorem 5.5. *For $n \geq 10$, G_n has no self-approaching drawing.*

The outline of the proof is as follows. We show that every self-approaching drawing Γ of G_{10} contains a self-approaching drawing of G_3 such that for each block μ of this G_3 , the angle at $r = r(\mu)$ is very small, angles at a and c are 90° or slightly larger (Lemma 5.9) and such that sides ra and rc have almost the same length which is significantly greater than $\text{dist}(a, c)$ (Lemma 5.11). In addition, the following properties hold for this G_3 .

1. If μ_i is contained in the subcactus rooted at c_j , each self-approaching b_i - a_j -path uses edge $b_i a_i$, and analogously for the symmetric case; see Lemma 5.10.
2. Each block is drawn much smaller than its parent block; see Lemma 5.12(i).

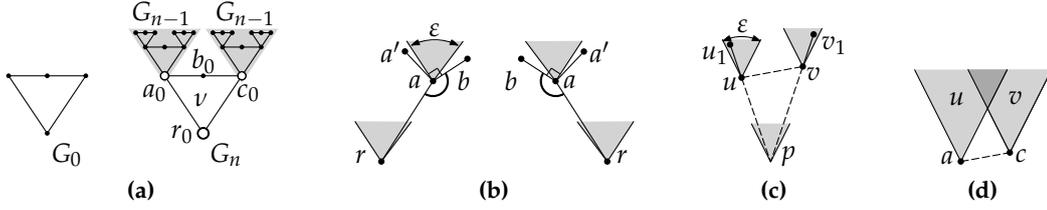


Figure 42: (a) cactuses G_n ; (b) Lemma 5.9(iii); (c),(d): Lemma 5.9(iv).

3. If the descendants of block μ form subcactuses G_k with $k \geq 2$ on both sides, the parent block of μ must be drawn smaller than μ ; see Lemma 5.12(ii).

Obviously, the second and third conditions are contradictory. Note that every block has to be self-approaching. However, it might be non-convex and even non-planar.

Observation 5.1. *In a self-approaching drawing of a polygon P , no two non-consecutive angles can be both less than 90° .*

Proof. If P is a triangle, it is trivially self-approaching. Let now v_1, v_2, v_3, v_4 be pairwise distinct vertices appearing in this circular order around the boundary of P . Let the angles at both v_2 and v_4 be less than 90° . However, a self-approaching v_1 - v_3 -path must use either v_2 or v_4 , a contradiction. \square

The following lemmas will be used to show that the drawings of certain blocks must be relatively thin, i.e., their downward edges have similar directions.

Lemma 5.8. *Every self-approaching drawing of G_{10} contains a cutvertex \bar{r} , such that $\text{depth}_C(\bar{r}) = 4$, and every pair of directions in $U_{\bar{r}}$ form an angle at most $\varepsilon = 22.5^\circ$.*

Proof. Denote by $r_j, j = 1, \dots, 16$, the cutvertices with $\text{depth}_C(r_j) = 4$. By an argument similar to the one in the proof of Lemma 5.4, the edges in U_{r_j} appear in the following circular order by their directions: first the edges in $U_{r_{\pi(1)}}$, then the edges in $U_{r_{\pi(2)}}, \dots$, then the edges in $U_{r_{\pi(16)}}$ for some permutation π . Therefore, by the pigeonhole principle, the statement holds for some $j \in \{1, \dots, 16\}$ and $\bar{r} = r_j$. \square

Let \bar{r} be a cutvertex in the fixed drawing at $\text{depth}_C(\bar{r}) = 4$ with the property shown in Lemma 5.8. Then, $G^{\bar{r}}$ is isomorphic to G_6 . From now on, we only consider non-leaf blocks μ_i and vertices r_i, a_i, b_i, c_i in $G^{\bar{r}}$. We shall sometimes name the points a instead of a_i etc. for convenience. We assume $\angle(\vec{e}_2, \vec{ra}), \angle(\vec{e}_2, \vec{rc}) \leq \varepsilon/2$, i.e., edges ra, rc are “almost vertical”. The following lemma is proved using basic trigonometric arguments.

Lemma 5.9. *The following facts hold.*

- (i) $\angle abc \geq 90^\circ$;
- (ii) $G^a \subseteq \mathbf{h}_b^a, G^c \subseteq \mathbf{h}_b^c$;

- (iii) $\angle bar \leq 90^\circ + \varepsilon, \angle bcr \leq 90^\circ + \varepsilon.$
- (iv) For vertices u in G^a, v in G^c of degree 4 we have $\angle(\vec{u\bar{v}}, \vec{e_1}) \leq \varepsilon/2.$

Proof. (i) We have $\angle arc \leq \varepsilon.$ Thus, by Observation 5.1, $\angle abc \geq 90^\circ.$

(ii) Let t be a vertex of $G^c.$ Since $\angle arc \leq \varepsilon < 90^\circ,$ any self-approaching a - t -path must contain $bc.$ Thus, $t \in h_b^c,$ and the claim for G^c and, similarly, for G^a follows.

(iii) Consider block μ' containing $a' \neq a, r(\mu') = a;$ see Figure 42b. Then, we have $\angle(\vec{r\bar{a}}, \vec{a\bar{a}'}) \leq \varepsilon.$ By (ii), $baa' \geq 90^\circ.$ If $\angle bar > 90^\circ + \varepsilon,$ then $\angle(\vec{r\bar{a}}, \vec{a\bar{a}'}) > \varepsilon,$ a contradiction. The same argument applies for $\angle bcr.$

(iv) Since u, v have degree 4, they are roots of some blocks. Let u_1 be a neighbor of u in G^u and v_1 a neighbor of v in G^v maximizing $\angle u_1uv$ and $\angle v_1vu;$ see Figure 42c. By considering self-approaching u_1 - v - and v_1 - u -paths, it follows $\angle u_1uv, \angle v_1vu \geq 90^\circ.$ Also, $\text{ray}(u_1, u)$ and $\text{ray}(v_1, v)$ converge by Lemma 5.2. Let p be their intersection. Then, $\angle upv \leq \varepsilon$ and $\angle puv, \angle pvu \leq 90^\circ.$ We have $\angle(\vec{p\bar{u}}, \vec{e_2}) \leq \varepsilon/2$ and $\angle(\vec{p\bar{v}}, \vec{e_2}) \leq \varepsilon/2.$ Therefore, if $\vec{u\bar{v}}$ points upward, it forms an angle at most $\varepsilon/2$ with the horizontal direction. If $\vec{u\bar{v}}$ points downward, by symmetric arguments, $\vec{v\bar{u}}$ forms an angle at most $\varepsilon/2$ with the horizontal direction. The same holds for $\vec{a\bar{c}}, \vec{a\bar{v}}, \vec{u\bar{c}}.$

It remains to show that u is “to the left” of $v.$ Since $\angle_{\text{ccw}}(\vec{r\bar{c}}, \vec{r\bar{a}}) < 180^\circ$ and $\angle(\vec{r\bar{c}}, \vec{r\bar{a}}) \leq \varepsilon,$ we have $\angle(\vec{a\bar{c}}, \vec{e_1}) \leq \varepsilon/2.$ Consider the two vertically aligned cones with apices a and c and angle ε (gray in Figure 42d). Vertex u must be in the cone of $a,$ and vertex v in the cone of $c.$ If u is not in the cone of c and, at the same time, v not in the cone of $a,$ then v is to the right of $u.$ In this case, we have $\angle(\vec{u\bar{v}}, \vec{e_1}) \leq \varepsilon/2,$ and we are done.

Now assume $\angle(\vec{u\bar{v}}, -\vec{e_1}) \leq \varepsilon/2.$ Then, by the above argument, u is in the cone of c or v in the cone of a (without loss of generality, u is in the cone of a). Thus, u must be in the dark gray area in Figure 42d). This contradicts the fact that $\vec{u\bar{c}}$ forms an angle of at most $\varepsilon/2$ with the horizontal direction. □

We can now describe block angles at a_i, c_i more precisely and characterize certain self-approaching paths in $G^{\bar{f}}.$ We show that a self-approaching path from b_i downwards and to the left, i.e., to an ancestor block μ_j of $\mu_i,$ such that μ_i is in $G^{c_j},$ must use $a_i.$ Similarly, a self-approaching path downwards and to the right must use $c_i.$ Since for several ancestor blocks of μ_i the roots lie on both of these two kinds of paths, we can bound the area containing them and show that it is relatively small. This implies that the ancestor blocks are small as well, providing a contradiction.

We say that point p lies to the left of $\text{ray}(u, v)$ if we have $0 \leq \angle_{\text{ccw}}(\vec{u\bar{v}}, \vec{u\bar{p}}) \leq 90^\circ.$ We say that p lies to the right of $\text{ray}(u, v)$ if we have $0 \leq \angle_{\text{ccw}}(\vec{u\bar{p}}, \vec{u\bar{v}}) \leq 90^\circ.$

left/right of ray

Lemma 5.10. Consider non-leaf blocks $\mu_0, \mu_1, \mu_2,$ such that $r(\mu_1) = c_0$ and μ_2 is in $G^{a_1};$ see Figure 43a.

- (i) We have $\angle r_2a_2b_2, \angle r_2c_2b_2 \in [90^\circ, 90^\circ + \varepsilon].$ Furthermore, b_2 lies to the right of $\text{ray}(r_2, a_2)$ and to the left of $\text{ray}(r_2, c_2).$

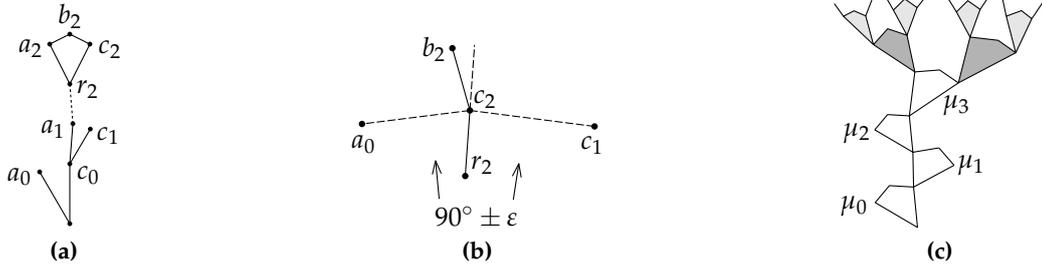


Figure 43: (a),(b) construction for Lemma 5.10; (c) subcactus G_6 providing the contradiction in the proof of Theorem 5.5.

(ii) Each self-approaching b_2 - a_0 -path uses a_2 ; each self-approaching b_2 - c_1 -path uses c_2 .

Proof. (i) Assume $\angle r_2 a_2 b_2 < 90^\circ$. Then, all self-approaching b_2 - a_0 - and b_2 - c_1 -paths must use c_2 . By Lemma 5.9(iv), the lines through $a_0 c_2$ and $c_2 c_1$ are “almost horizontal”, i.e., $\angle(\overrightarrow{a_0 c_2}, \vec{e}_1), \angle(\overrightarrow{c_2 c_1}, \vec{e}_1) \leq \varepsilon/2$. Since $r_2 c_2$ is “almost vertical”, r_2 must lie below these lines, and $\angle a_0 c_2 r_2, \angle c_1 c_2 r_2 \in [90^\circ - \varepsilon, 90^\circ + \varepsilon]$; see Figure 43b.

First, let b_2 lie to the left of $\text{ray}(r_2, c_2)$. Recall that by our assumption, $\angle r_2 c_2 b_2 \geq 90^\circ$. Furthermore, since every self-approaching b_2 - a_0 -path must use c_2 , we have $\angle a_0 c_2 b_2 \geq 90^\circ$. Therefore, b_2 cannot lie inside the counterclockwise angle between $c_2 a_0$ and $c_2 r_2$, since $\angle_{\text{ccw}}(\overrightarrow{c_2 a_0}, \overrightarrow{c_2 r_2}) \leq 90^\circ + \varepsilon < \angle r_2 c_2 b_2 + \angle a_0 c_2 b_2$. Thus, b_2 is above $a_0 c_2$, and we have $\angle r_2 c_2 b_2 = \angle a_0 c_2 r_2 + \angle a_0 c_2 b_2 \geq (90^\circ - \varepsilon) + 90^\circ = 180^\circ - \varepsilon$. Since $\varepsilon < 22.5^\circ$, this contradicts Lemma 5.9(iii).

Now let b_2 lie to the right of $\text{ray}(r_2, c_2)$. Recall that every self-approaching b_2 - c_1 -path must use c_2 , so $\angle c_1 c_2 b_2 \geq 90^\circ$. Therefore, b_2 cannot lie inside the counterclockwise angle between $c_2 r_2$ and $c_2 c_1$, since $\angle_{\text{ccw}}(\overrightarrow{c_2 r_2}, \overrightarrow{c_2 c_1}) \leq 90^\circ + \varepsilon < \angle r_2 c_2 b_2 + \angle b_2 c_2 c_1$. Thus, b_2 is above $c_2 c_1$, and $\angle r_2 c_2 b_2 = \angle c_1 c_2 r_2 + \angle c_1 c_2 b_2 \geq (90^\circ - \varepsilon) + 90^\circ = 180^\circ - \varepsilon$. Again, since $\varepsilon < 22.5^\circ$, this contradicts Lemma 5.9(iii). It follows $\angle r_2 a_2 b_2 \geq 90^\circ$.

Analogously, we prove $\angle r_2 c_2 b_2 \geq 90^\circ$. Thus, by Lemma 5.9(iii), we have $\angle r_2 a_2 b_2, \angle r_2 c_2 b_2 \in [90^\circ, 90^\circ + \varepsilon]$. Since $\angle a_2 b_2 c_2 \geq 90^\circ$ by Lemma 5.9(i), b_2 lies to the right of $\text{ray}(r_2, a_2)$ and to the left of $\text{ray}(r_2, c_2)$. (If b_2 lies to the left of both rays, then $\angle a_2 b_2 c_2 = \angle(\overrightarrow{a_2 b_2}, \overrightarrow{c_2 b_2}) \leq 2\varepsilon < 90^\circ$.)

(ii) Similarly, if a self-approaching b_2 - a_0 -path uses c_2 instead of a_2 , then $\angle r_2 c_2 b_2 \geq 180^\circ - \varepsilon$. The last part follows analogously. \square

The next lemma allows us to show that certain blocks are drawn smaller than their ancestors.

Lemma 5.11. *The following facts hold.*

(i) $\frac{|ra|}{|rc|}, \frac{|rc|}{|ra|} \geq \cos \varepsilon;$

(ii) $\frac{|ac|}{|ra|}, \frac{|ac|}{|rc|} \leq \tan \varepsilon;$

(iii) *The distance from a to the line through rc is at least $|ac| \cos \varepsilon$.*

(iv) Consider block μ containing a, b, c, r , vertex $u \neq a$ in G^a and $v \neq c$ in G^c , $\deg(u) = \deg(v) = 4$. Then, $\frac{|au|}{|ac|} \leq \tan \varepsilon$, $\frac{|cv|}{|ac|} \leq \tan \varepsilon$, and $|uv| \leq (1 + 2 \tan \varepsilon)|ac|$.

Proof. (i) Due to symmetry, we show only one part. By Lemma 5.9(iv), $\angle acr \in [90^\circ - \varepsilon, 90^\circ + \varepsilon]$. Therefore,

$$\frac{|ra|}{|rc|} = \frac{\sin \angle acr}{\sin \angle rac} \geq \frac{\sin(90^\circ - \varepsilon)}{1} = \cos \varepsilon.$$

(ii) We have $\angle arc \leq \varepsilon$. Therefore,

$$\frac{|ac|}{|ra|} = \frac{\sin \angle arc}{\sin \angle acr} \leq \frac{\sin(\varepsilon)}{\sin(90^\circ - \varepsilon)} = \tan \varepsilon.$$

(iii) Let d be the point on the line through rc minimizing $|ad|$. Since $\angle acr \in [90^\circ - \varepsilon, 90^\circ + \varepsilon]$, we have $\angle(\vec{ac}, \vec{ad}) \leq \varepsilon$. Thus, $|ad| \geq |ac| \cos \varepsilon$.

(iv) By Lemma 5.9(iv), $\angle acu \leq \varepsilon$ and $\angle auc \in [90^\circ - \varepsilon, 90^\circ + \varepsilon]$. Thus,

$$\frac{|au|}{|ac|} = \frac{\sin \angle acu}{\sin \angle auc} \leq \frac{\sin \varepsilon}{\sin(90^\circ - \varepsilon)} = \tan \varepsilon.$$

Similarly, $|vc| \leq |ac| \tan \varepsilon$. Thus, $|uv| \leq |ua| + |ac| + |cv| \leq (1 + 2 \tan \varepsilon)|ac|$. \square

From now on, let μ_0 be the root block of $G^{\bar{r}}$ and μ_1, μ_2, μ_3 its descendants such that $r(\mu_1) = c_0, r(\mu_2) = a_1, r(\mu_3) \in \{a_2, c_2\}$; see Figure 43c. Light gray blocks are the subject of Lemma 5.12(i), which shows that several ancestor roots lie inside a cone with a small angle. Dark gray blocks are the subject of Lemma 5.12(ii), which considers the intersection of the cones corresponding to a pair of sibling blocks and shows that some of their ancestor roots lie inside a narrow strip; see Figure 44a for a sketch.

Lemma 5.12. *Let μ be a block in G^{c_2} with vertices $a, b, c, r(\mu)$.*

(i) *Let μ have depth 5 in $G^{\bar{r}}$. Then, the cone $\mathbf{h}_b^a \cap \mathbf{h}_b^c$ contains $r(\mu), r(\pi(\mu)), r(\pi^2(\mu))$ and $r(\pi^3(\mu))$.*

(ii) *Let μ have depth 4 in $G^{\bar{r}}$. There exist u in G^a and v in G^c of degree 4 and a strip S containing $r(\mu), r(\pi(\mu)), r(\pi^2(\mu)) = r(\mu_2)$, such that u and v lie on the different boundaries of S , and it holds: $|uv| \leq (1 + 2 \tan \varepsilon)(\tan \varepsilon) \min\{|r(\mu)a|, |r(\mu)c|\}$.*

Proof. (i) Consider a self-approaching b - b_0 -path ρ_0 and a self-approaching b - b_1 -path ρ_1 . By Lemma 5.10(ii) applied to μ , ba is the first edge of ρ_0 and bc the first edge of ρ_1 . Since the cutvertices $r(\mu), r(\pi(\mu)), r(\pi^2(\mu)), r(\pi^3(\mu))$ are on both ρ_0 and ρ_1 , the statement holds.

(ii) Consider blocks μ_l, μ_r , such that $r(\mu_l) = a$ and $r(\mu_r) = c$. By (i), $r(\mu), r(\pi(\mu)), r(\pi^2(\mu))$ are in $\Lambda := \mathbf{h}_{b_l}^a \cap \mathbf{h}_{b_l}^{c_l} \cap \mathbf{h}_{b_r}^{a_r} \cap \mathbf{h}_{b_r}^{c_r}$. Let \vec{v}_l be the vector $\vec{b_l c_l}$ rotated by 90° clockwise and \vec{v}_r be the vector $\vec{b_r a_r}$ rotated by 90° counterclockwise. Note that by Lemma 5.9(ii), G^{c_l}, G^{a_r} lie in $\mathbf{h}_{b_l}^{c_l}, \mathbf{h}_{b_r}^{a_r}$ respectively. Therefore, $\text{ray}(c_l, \vec{v}_l)$

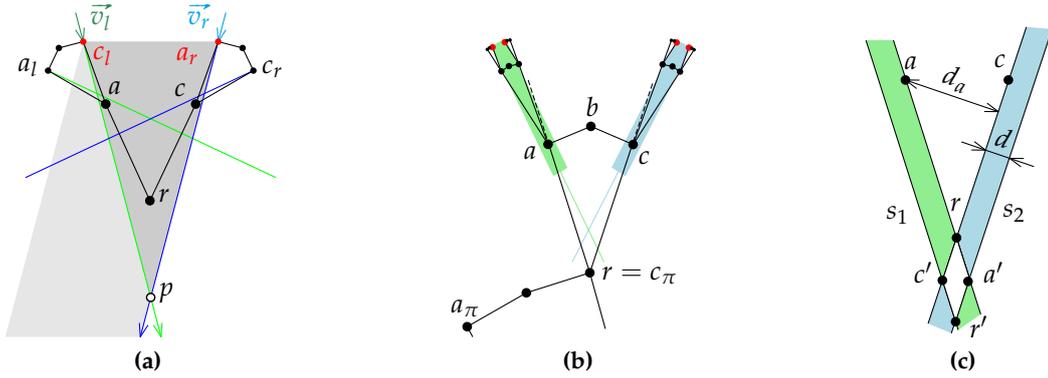


Figure 44: Showing the contradiction in Theorem 5.5.

and $\text{ray}(a_r, \vec{v}_r)$ (green resp. blue arrows in Figure 44a) converge, since the converse would contradict Lemma 5.2. Let p be their intersection. Due to the chosen directions, $r(\mu)$, $r(\pi(\mu))$, $r(\pi^2(\mu))$ are below both c_l and a_r . Therefore, $r(\mu)$, $r(\pi(\mu))$, $r(\pi^2(\mu))$ are contained in the triangle $c_l a_r p$, which lies inside a strip S of width at most $|c_l a_r|$, whose respective boundaries contain c_l and a_r . By Lemma 5.11(iv) and (ii), $|c_l a_r| \leq (1 + 2 \tan \varepsilon) |ac| \leq (1 + 2 \tan \varepsilon) (\tan \varepsilon) \min\{|r(\mu)a|, |r(\mu)c|\}$. \square

Again, we consider two siblings and the intersection of their corresponding strips, which forms a small diamond containing the root of the ancestor block; see Figure 44b, 44c.

Lemma 5.13. Consider block μ_3 containing $r = r(\mu_3)$, a, b, c , and let $r_\pi := r(\pi(\mu_3))$. The following facts hold.

- (i) $|rr_\pi| \leq \frac{(1+2 \tan \varepsilon)(\tan \varepsilon)^2}{\cos \varepsilon} (|ra| + |rc|)$;
- (ii) $|ra|, |rc| \leq |rr_\pi| (\tan \varepsilon)^2$.

Proof. (i) Define $d = (1 + 2 \tan \varepsilon) (\tan \varepsilon)^2 |ac|$. Then, by Lemma 5.11(ii) and (iv) and Lemma 5.12(ii), vertices a, r and r_π are contained in a strip s_1 (green in Figure 44b) of width at most d . Additionally, both boundaries of s_1 contain vertices of G^a (red dots), which lie in \mathbf{h}_b^a and, by Lemma 5.10(i), to the left of $\text{ray}(r, a)$. Thus, the downward direction along s_1 is counterclockwise compared to $\vec{a}\vec{r}$. (Otherwise, the green strip could not contain a .) Similarly, vertices c, r and r_0 are contained in a strip s_2 (blue) of width at most d , and both boundaries of s_2 contain vertices of G^c , which lie to the right of $\text{ray}(r, c)$. Thus, the downward direction along s_2 is clockwise compared to $\vec{c}\vec{r}$; see Figure 44b.

Let us find an upper bound for the diameter of the parallelogram $s_1 \cap s_2$. In the critical case, the right side of s_1 touches ra , the left side of s_2 touches rc , and the width of both strips is d ; see Figure 44c. Let a' (resp. c') be the intersection of the right (resp. left) sides of s_1 and s_2 , and r' the intersection of the left side of s_1 and right side of s_2 . Let d_a be the distance from a to the line through rc and d_c the distance from c to the line through ra . By Lemma 5.11(iii), we

have $d_a, d_c \geq |ac| \cos \varepsilon$. Moreover, $\frac{|ra'|}{|ra|} = \frac{d}{d_a}$ and $\frac{|rc'|}{|rc|} = \frac{d}{d_c}$. Therefore, $|ra'| \leq \frac{d|ra|}{|ac| \cos \varepsilon}$, $|rc'| \leq \frac{d|rc|}{|ac| \cos \varepsilon}$ and

$$|rr'| \leq |ra'| + |rc'| \leq \frac{(1 + 2 \tan \varepsilon)(\tan \varepsilon)^2}{\cos \varepsilon} (|ra| + |rc|).$$

Since $\angle a'rc' \leq \varepsilon$, rr' is the diameter, thus, $|rr_\pi| \leq |rr'|$.

(ii) Let a_π and c_π be the two neighbors of r_π in the block $\mu_2 = \pi(\mu_3)$. We have $r \in \{a_\pi, c_\pi\}$. Assume $r = c_\pi$ as in Figure 44b. By Lemma 5.11(iv), $\frac{|ac_\pi|}{|a_\pi c_\pi|} \leq \tan \varepsilon$. By Lemma 5.11(ii), $\frac{|a_\pi c_\pi|}{|r_\pi c_\pi|} \leq \tan \varepsilon$. It follows: $|ra| = |ac_\pi| \leq |r_\pi c_\pi| (\tan \varepsilon)^2 = |rr_\pi| (\tan \varepsilon)^2$. Analogously, $|rc| \leq |rr_\pi| (\tan \varepsilon)^2$. \square

For $\varepsilon \leq 22.5^\circ$, the two claims of Lemma 5.13 contradict each other. This concludes the proof of Theorem 5.5.

5.4 PLANAR INCREASING-CHORD DRAWINGS OF 3-TREES

In this section, we show how to construct planar increasing-chord drawings of planar 3-trees. We make use of *Schnyder labelings* [Sch90] and drawings of triangulations based on them. For a plane triangulation $G = (V, E)$ with external vertices r, g, b , its Schnyder labeling is an orientation and partition of the interior edges into three trees T_r, T_g, T_b (called *red*, *green* and *blue tree*), such that for each internal vertex v , its incident edges appear in the following clockwise order: exactly one outgoing red, an arbitrary number of incoming blue, exactly one outgoing green, an arbitrary number of incoming red, exactly one outgoing blue, an arbitrary number of incoming green. Each of the three outer vertices r, g, b serves as the root of the tree in the same color and all its incident interior edges are incoming in the respective color. For $v \in V$, let R_v^r (the *red region* of v) denote the region bounded by the v - g -path in T_g , the v - b -path in T_b and the edge gb . Let $|R_v^r|$ denote the number of the interior faces in R_v^r . The green and blue regions R_v^g, R_v^b are defined analogously. Assigning v the coordinates $(|R_v^r|, |R_v^g|, |R_v^b|) \in \mathbb{R}^3$ results in a plane straight-line drawing of G in the plane $\{x = (x_1, x_2, x_3) \mid x_1 + x_2 + x_3 = f - 1\}$ called *Schnyder drawing*. Here, f denotes the number of faces of G . For a thorough introduction to this topic, see the book of Felsner [Felo4].

Schnyder labeling

regions R_v^r, R_v^g, R_v^b

Schnyder drawing

For $\alpha, \beta \in [0^\circ, 360^\circ]$, let $[\alpha, \beta]$ denote the corresponding counterclockwise cone of directions. We consider drawings satisfying the following constraints.

Definition 5.1. Let $G = (V, E)$ be a plane triangulated graph with a Schnyder labeling. For $0^\circ \leq \alpha \leq 60^\circ$, we call an arbitrary planar straight-line drawing of G α -Schnyder if for each internal vertex $v \in V$, its outgoing red edge has direction in $[90^\circ - \frac{\alpha}{2}, 90^\circ + \frac{\alpha}{2}]$, blue in $[210^\circ - \frac{\alpha}{2}, 210^\circ + \frac{\alpha}{2}]$ and green in $[330^\circ - \frac{\alpha}{2}, 330^\circ + \frac{\alpha}{2}]$ (see Figure 45a).

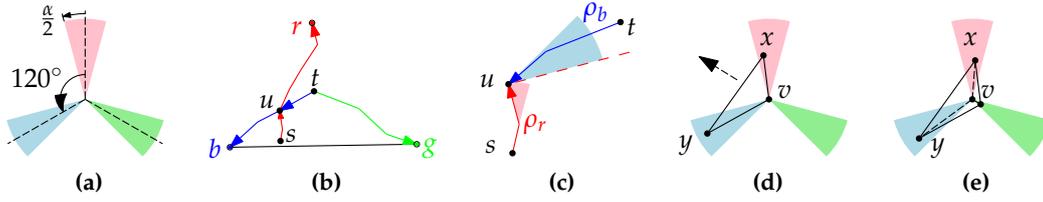


Figure 45: (a)–(c) 30° -Schnyder drawings are increasing-chord; (d),(e) special case of planar 3-trees.

According to Definition 5.1, classical Schnyder drawings are 60° -Schnyder; see, e.g., Lemma 4 in [Dha10]. The next lemma shows an interesting connection between α -Schnyder and increasing-chord drawings.

Lemma 5.14. *For any $\alpha \leq 30^\circ$, α -Schnyder drawings are increasing-chord drawings.*

Proof. Let $G = (V, E)$ be a plane triangulation with a given Schnyder labeling and Γ a corresponding 30° -Schnyder drawing. Let r, g, b be the red, green and blue external vertex, respectively, and T_r, T_g, T_b the directed trees of the corresponding color.

Consider vertices $s, t \in V$. First, note that monochromatic directed paths in Γ have increasing chords by Lemma 5.1. Assume s and t are not connected by such a path. Then, they are both internal and s is contained in one of the regions R_t^r, R_t^g, R_t^b . Without loss of generality, we assume $s \in R_t^r$. The s - r -path in T_r crosses the boundary of R_t^r , and we assume without loss of generality that it crosses the blue boundary of R_t^r in $u \neq t$; see Figure 45b. The other cases are symmetric.

Let ρ_r be the s - u -path in T_r and ρ_b the t - u -path in T_b ; see Figure 45c. On the one hand, the direction of a line orthogonal to a segment of ρ_r is in $[345^\circ, 15^\circ] \cup [165^\circ, 195^\circ]$. On the other hand, ρ_b is contained in a cone $[15^\circ, 45^\circ]$ with apex u . Thus, $\rho_b^{-1} \subseteq \text{front}(\rho_r)$, and $\rho_r \cdot \rho_b^{-1}$ is self-approaching by Fact 5.2. By a symmetric argument it is also self-approaching in the other direction, and hence has increasing chords. \square

planar 3-tree

Planar 3-trees are the graphs that can be obtained from a triangle by repeatedly choosing a (triangular) face f , inserting a new vertex v into f , and connecting v to each vertex of f .

Lemma 5.15. *Planar 3-trees have α -Schnyder drawings for any $0^\circ < \alpha \leq 60^\circ$.*

Proof. We describe a recursive construction of an α -Schnyder drawing of a planar 3-tree. We use the pattern in Figure 45a consisting of three cones with angle $0^\circ < \alpha \leq 60^\circ$ to maintain the following invariant.

For each inner face f , the pattern can be centered at a point p in the interior of f , such that every cone of the pattern contains one vertex of f in its interior.

We start with an equilateral triangle. Obviously, the invariant holds for the single inner face f by choosing p to be the barycenter of f .

Assume the invariant holds for each inner face of the drawing created so far. We prove that the invariant can be maintained after adding a new vertex. Consider an inner face f with corners x, y, z . We move the pattern from Figure 45a, such that its center lies in the interior point p of f from the invariant. Without loss of generality, let x be in the red cone of the pattern, y in the blue cone and z in the green. We insert the new vertex v at point p and connect v to x, y, z . We make the edge vx outgoing red, vy outgoing blue and vz outgoing green.

We now show that the invariant holds for the three newly created faces $f_1 = xyv$, $f_2 = yzv$ and $f_3 = zxv$. Consider f_1 first. If we place the pattern at v , by the invariant for face f , one cone of the pattern contains x and another contains y in its interior; see Figure 45d. It is now possible to move the pattern inside the triangle xyv slightly, such that v is in the interior of the third cone; see Figure 45e. This proves the invariant for f_1 , and the proof for f_2 and f_3 is analogous. \square

Lemmas 5.14 and 5.15 provide a constructive proof for the following theorem.

Theorem 5.6. *Every planar 3-tree has a planar increasing-chord drawing.*

5.5 SELF-APPROACHING DRAWINGS IN THE HYPERBOLIC PLANE

Kleinberg [Kle07] showed that every tree can be drawn greedily in the hyperbolic plane \mathbb{H}^2 . This is not the case in \mathbb{R}^2 . Thus, \mathbb{H}^2 is more powerful than \mathbb{R}^2 in this regard. Since self-approaching drawings are closely related to greedy drawings, it is natural to investigate the existence of self-approaching drawings in \mathbb{H}^2 .

We shall use the *Poincaré disk* model for \mathbb{H}^2 , in which \mathbb{H}^2 is represented by the unit disk $D = \{x \in \mathbb{R}^2 : |x| < 1\}$ and the geodesics are represented by arcs of circles orthogonal to the boundary of D . We consider a drawing of a graph in \mathbb{H}^2 straight-line, if the edges are drawn as arcs of such circles. For an introduction to the Poincaré disk model, see, for example, Kleinberg [Kle07] and the references therein.

Poincaré disk D

First, let us consider a tree $T = (V, E)$. A drawing of T in \mathbb{R}^2 is self-approaching if and only if no normal on an edge of T in any point crosses another edge [Ala+13]. The same condition holds in \mathbb{H}^2 .

Lemma 5.16. *A straight-line drawing Γ of a tree T in \mathbb{H}^2 is self-approaching if and only if no normal on an edge of T crosses Γ in another point.*

Proof. The proof is similar to the Euclidean case. We present it for the sake of completeness. First, let Γ be a self-approaching drawing, for which the condition of the lemma is violated. Without loss of generality, let $\rho = (s, u, \dots, t)$ be the st -path in T , such that the normal on su in a point r crosses ρ in another point. Due to the piecewise linearity of ρ , we may assume r to be in the interior of su . Let $H_+ = \{p = (p_x, p_y) \in D \mid p_y > 0\}$ and $H_- = \{p = (p_x, p_y) \in D \mid p_y < 0\}$ the top and bottom hemispheres of D . For $p_1, p_2 \in D$, let $d(p_1, p_2)$ denote the

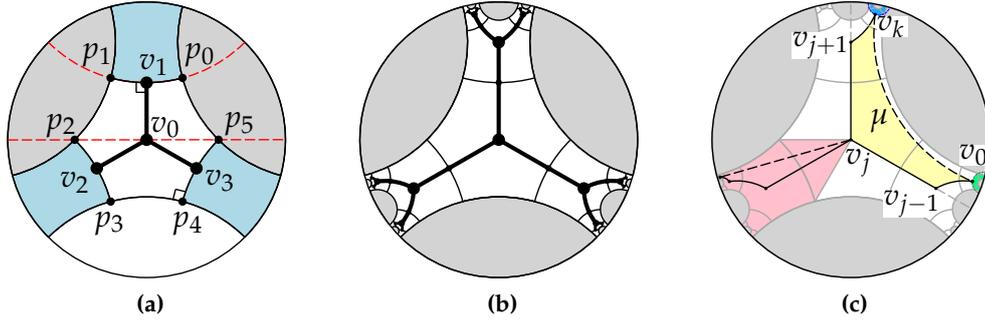


Figure 46: Constructing increasing-chord drawings of binary trees and cactuses in \mathbb{H}^2 .

hyperbolic distance between p_1 and p_2 , i.e., the hyperbolic length of the corresponding geodesics. We recall the following basic fact whose proof is given, e.g., by Kleinberg [Kle07].

Claim 1. Let $0 < y < 1$, $p_- = (0, -y)$, $p_+ = (0, y)$. Then, for each $p \in H_-$, $d(p, p_-) < d(p, p_+)$.

Due to isometries, we can assume that r is in the origin of D , su is vertical, $s \in H_-$, $u \in H_+$. Let $a \in H_-$, $b \in H_+$ be two points on su , such that $|ar| = |rb|$. Since the normal on su in r crosses ρ , there must exist a point c on ρ , $c \in H_-$, such that a, b, c are on ρ in this order. However, $d(a, c) < d(b, c)$, a contradiction to ρ being self-approaching.

Let Γ be a drawing of T , for which the condition holds. Let a, b, c be three consecutive points on a path ρ in Γ . First, assume a, b lie on the same arc of Γ . We apply an isometry to Γ , such that ab is vertical, $a \in H_-$, $b \in H_+$, and a, b are equidistant from the origin o . The normal to ρ in o is the equator. Thus, $c \notin H_-$, and $d(b, c) \leq d(a, c)$. By applying this argument iteratively, this inequality also holds if a, b lie on different arcs. \square

According to the characterization by Alamdari et al. [Ala+13], some binary trees have no self-approaching drawings in \mathbb{R}^2 . We show that this is no longer the case in \mathbb{H}^2 .

Theorem 5.7. Let $T = (V, E)$ be a tree, such that each node of T has degree either 1 or 3. Then, T has a self-approaching drawing in \mathbb{H}^2 , in which every arc has the same hyperbolic length and every pair of incident arcs forms an angle of 120° .

Proof. For convenience, we subdivide each edge of T once. We shall show that both pieces are collinear in the resulting drawing Γ and have the same hyperbolic length.

First, consider a regular hexagon $\diamond = p_0p_1p_2p_3p_4p_5$ centered at the origin o of D ; see Figure 46a. In \mathbb{H}^2 , it can have angles smaller than 120° . We choose them to be 90° (any angle between 0° and 90° would work). Next, we draw a $K_{1,3}$ with center v_0 in o and the leaves v_1, v_2, v_3 in the middle of the arcs p_0p_1 , p_2p_3 , p_4p_5 respectively.

For each such building block of the drawing consisting of a $K_{1,3}$ inside a regular hexagon with 90° angles, we add its copy mirrored at an arc of the hexagon containing a leaf node of the tree constructed so far. For example, in the first iteration, we add three copies of \diamond mirrored at p_0p_1 , p_2p_3 and p_4p_5 , respectively, and the corresponding inscribed $K_{1,3}$ subtrees. The construction after two iterations is shown in Figure 46b. This process can be continued infinitely to construct a drawing Γ_∞ of the infinite binary tree. However, we stop after we have completed Γ for the tree T .

We now show that Γ_∞ (and thus also Γ) has the desired properties. Due to isometries and Lemma 5.16, it suffices to consider edge $e = v_0v_1$ and show that a normal on e does not cross Γ_∞ in another point. To see this, consider Figure 46a. Due to the choice of the angles of \diamond , all the other hexagonal tiles of Γ_∞ are contained in one of the three blue quadrangular regions $\square_i := \mathbf{h}_{v_0}^{v_i} \setminus (\mathbf{h}_{v_i}^{p_{2i-1}} \cup \mathbf{h}_{v_i}^{p_{2i-2}})$, $i = 1, 2, 3$. Thus, the regions $\mathbf{h}_{v_1}^{p_1}$ and $\mathbf{h}_{v_1}^{p_0}$ (gray) contain no point of Γ_∞ . Therefore, since each normal on v_0v_1 is contained in the “slab” $D \setminus (\mathbf{h}_{v_0}^{v_1} \cup \mathbf{h}_{v_1}^{v_0})$ bounded by the diameter through p_2, p_5 and the line through p_0, p_1 (dashed) and is parallel to both of these lines, it contains no other point of Γ_∞ . \square

We note that our proof is similar in spirit to the one by Kleinberg [Kle07], who also used tilings of \mathbb{H}^2 to prove that any tree has a greedy drawing in \mathbb{H}^2 .

As in the Euclidean case, it can be easily shown that if a tree T contains a node v of degree 4, it has a self-approaching drawing in \mathbb{H}^2 if and only if T is a subdivision of $K_{1,4}$ (apply an isometry, such that v is in the origin of D). This completely characterizes the trees admitting a self-approaching drawing in \mathbb{H}^2 . Further, it is known that every binary cactus and, therefore, every 3-connected planar graph has a binary spanning tree [AFG10; LM10].

Corollary 5.2. (i) *A tree T has an increasing-chord drawing in \mathbb{H}^2 if and only if T either has maximum degree 3 or is a subdivision of $K_{1,4}$.*

(ii) *Every binary cactus and, therefore, every 3-connected planar graph has an increasing-chord drawing in \mathbb{H}^2 .*

Again, note that this is not the case for binary cactuses in \mathbb{R}^2 ; see the example in Theorem 5.5. We use the above construction to produce *planar* self-approaching drawings of binary cactuses in \mathbb{H}^2 . We show how to choose a spanning tree and angles at vertices of degree 2, such that non-tree edges can be added without introducing crossings; see Figure 46c for a sketch.

Corollary 5.3. *Every binary cactus has a planar increasing-chord drawing in \mathbb{H}^2 .*

Proof. Without loss of generality, let G be a binary cactus rooted at block v such that each block μ of G is either a single edge or a cycle. For each block μ forming a cycle $r(\mu) = v_0, v_1, \dots, v_k, v_0$, we remove edge v_0v_k , thus obtaining a binary tree T . We embed it similarly to the proof of Theorem 5.7 such that additionally the counterclockwise angle $\angle v_{j-1}v_jv_{j+1} = 120^\circ$ for $j = 1, \dots, k-1$. Obviously, T

is drawn in a planar way since for each edge e of T , each half of e is drawn inside its hexagon.

It remains to show that for each μ , adding arc v_0v_k introduces no crossings. For each $j = 1, \dots, k - 1$, we can apply an isometry to the drawing, such that v_j is in the origin and $\overrightarrow{v_jv_{j+1}}$ points upwards; see Figure 46c. According to the construction of T , subcactus $G_\mu^{v_0}$ (maximal subcactus of G containing v_0 and no other vertex of μ) lies in the green region contained in $\mathbf{h}_{v_1}^{v_0}$ and $G_\mu^{v_k}$ in the blue region contained in $\mathbf{h}_{v_{k-1}}^{v_k}$. Since $v_0 \notin \mathbf{h}_{v_{k-1}}^{v_k}$ and $v_k \notin \mathbf{h}_{v_1}^{v_0}$, arc v_0v_k crosses neither $G_\mu^{v_0}$ nor $G_\mu^{v_k}$. Furthermore, v_0 and v_k lie inside the 120° cone Λ_j formed by $\text{ray}(v_j, v_{j+1})$ and $\text{ray}(v_j, v_{j-1})$. Thus, v_0v_k does not cross $v_{j-1}v_j, v_jv_{j+1}$. Since subcactus $G_\mu^{v_j}$ is in $\mathbb{H}^2 \setminus \Lambda_j$ (it lies in the red area in Figure 46c), it is not crossed by v_0v_k either. \square

5.6 BOUNDED DILATION FOR EUCLIDEAN GREEDY DRAWINGS OF CACTUSES

In this section, we prove that Euclidean greedy drawings of trees and cactuses have bounded dilation. To prove this, we use *generalized self-approaching curves* introduced by Aichholzer et al. [Aic+01].

Definition 5.2 ([Aic+01]). *For an angle $\varphi \in [0, 180^\circ)$, an oriented curve is φ -self-approaching, if for any point b on the curve, the rest of the curve lies inside a wedge of angle φ with apex in b .*

The standard self-approaching curves studied by Icking et al. [IKL99] are 90° -self-approaching. Aichholzer et al. [Aic+01] proved a bound $c(\varphi)$ on the detour of every φ -self-approaching curve, where $c(\varphi)$ depends only on φ .

Lemma 5.17. *Let G be a path with endpoints s and t . A greedy drawing Γ of G is a curve that is 120° -self-approaching in both directions and, therefore, has detour $O(1)$ between s and t .*

Proof. Let ρ be the st -path in G . Consider an arbitrary vertex a of ρ , and let vertex b be the immediate successor of a on ρ . We show that for any point p on the edge ab , the part of ρ from p to t is contained in a 120° wedge with apex in p . For $b = t$, this is obviously the case. For $b \neq t$, consider the minimum wedge Λ with apex in b that contains the b - t -subpath of ρ . Let c, d be vertices defining

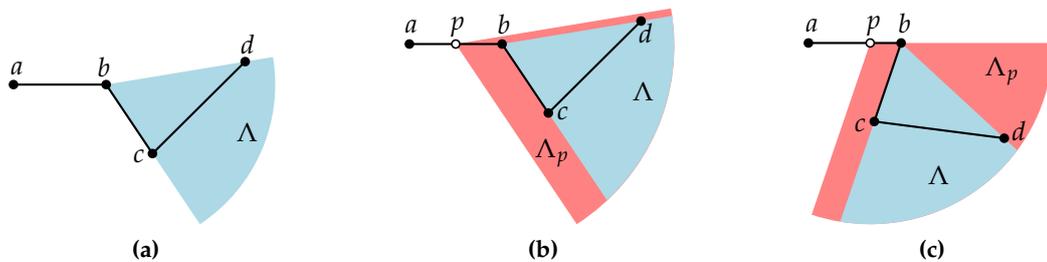


Figure 47: Proof of Lemma 5.17.

the two boundaries of Λ . Let $c \neq d$ (the case $c = d$ is analogous). Without loss of generality, let d come after c on ρ , let \vec{ab} be horizontal and point to the right, and let the counterclockwise order of vertices $\{a, c, d\}$ around b be a, c, d ; see Figure 47a. Due to the greediness of Γ , $|cd| < |bd|$. Therefore, the angle $\angle cbd$, which is the angle of Λ , is less than 90° .

Now let $p \neq b$. Consider the following two cases.

(1) Points c and d are on different sides of the line through ab . Let Λ_p be the cone with apex p and boundary rays $\text{ray}(p, \vec{bc})$ and $\text{ray}(p, \vec{bd})$, such that $p \in \Lambda_p$; see Figure 47b. Then, the angle of Λ_p is less than 90° , and $\Lambda \subseteq \Lambda_p$.

(2) Points c and d are not on different sides of the line through ab . Without loss of generality, let both points be on that line or below it. Recall that by greediness, $\angle abc > 60^\circ$. Let Λ_p be the cone with apex p and boundary rays $\text{ray}(p, \vec{bc})$ and $\text{ray}(p, \vec{ab})$, such that $c, d \in \Lambda_p$; see Figure 47c. The angle of Λ is less than 120° , and we have $\Lambda \subseteq \Lambda_p$.

This shows that the part of ρ from p to t is contained in a 120° wedge with apex in p . \square

Recall the definition of *weak* spanners by Schindelbauer et al. [SVZo7]. For some constant c and for every pair of vertices s, t , in a weak c -spanner there exists an st -path that remains within a circle around s with radius $c|st|$. Obviously, greedy drawings are weak 2-spanners. In contrast to Lemma 5.17, there exists a family of paths that are weak $(\sqrt{3} + 1/2)$ -spanners but do not have dilation $O(1)$ [SVZo7].

For every pair of vertices s, t in a greedy tree drawing Γ , the unique st -path is a greedy drawing, which implies the following corollary.

Corollary 5.4. *Every greedy drawing of a tree has dilation $O(1)$.*

Lemma 5.18. *Every greedy drawing of a cycle has dilation $O(1)$.*

Proof. Consider a greedy drawing Γ of a cycle and vertices s, t of Γ . If one of the two st -paths in Γ is a greedy drawing, the claim follows by Lemma 5.17. Now assume this is not the case. Let us note the following simple fact.

Fact. *A distance-decreasing st -path can not contain vertices u, v with $|uv| \geq 2|st|$.*

Let ρ be a distance-decreasing st -path in Γ . Path ρ lies inside a circle with radius $|st|$ and center t . We cover this circle by $c_t = O(1)$ tiles with diameter $\frac{1}{4}|st|$, e.g., regular hexagons; see Figure 48.

For a tile \mathcal{C}_i containing two or more vertices of ρ , let s_i be the first and t_i the last vertex of ρ in \mathcal{C}_i , and let ρ_i be the s_i - t_i -subpath of ρ . There are two simple s_i - t_i -paths in Γ ; one of them is ρ_i , and the other one contains both s and t . Only ρ_i is distance-decreasing by the above fact, since $|s_i t_i| \leq \frac{1}{4}|st|$. Therefore, for every pair of vertices u, v on ρ_i , $|uv| \leq \frac{1}{2}|st|$. Again, by the above fact, a distance-decreasing u - v -path can not contain both s and t . Therefore, the u - v -subpath of ρ_i is distance-decreasing in both directions, which shows that ρ_i is a greedy drawing.

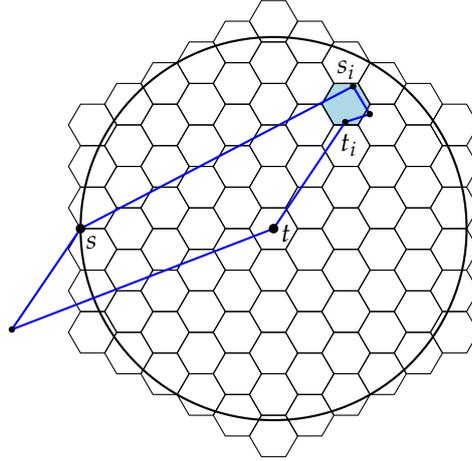


Figure 48: Proof of Lemma 5.18.

Let c be the constant bound from Lemma 5.17. A subpath of ρ from the first to the last vertex of a tile has length at most $\frac{c}{4}|st|$, and the following edge of ρ has length at most $2|st|$. Therefore, the total length of ρ is at most $c_t \cdot \frac{c}{4}|st| + (c_t - 1) \cdot 2|st| = O(|st|)$. \square

Theorem 5.8. *Every greedy drawing of a cactus has dilation $O(1)$.*

Proof. Consider a greedy drawing Γ of a cactus. In Γ , a distance-decreasing path between two vertices of the same block can not leave that block. Therefore, the drawing of every block in Γ is a greedy drawing of a cycle and has dilation $O(1)$ by Lemma 5.18.

For vertices s, t of Γ from different blocks, let v_1, \dots, v_k be the cutvertices visited by any simple st -path in this order. Then, by Lemma 5.2, the path $(v_0 = s, v_1, \dots, v_k, v_{k+1} = t)$ is a greedy drawing. Let c_1 be the constant bound from Lemma 5.17 and c_2 the constant bound from Lemma 5.18. By Lemma 5.18, the length of the shortest v_i - v_{i+1} -path is at most $c_2|v_i v_{i+1}|$ for $i = 0, \dots, k$. By Lemma 5.17, concatenating such paths creates an st -path of length at most $c_2 \sum_{i=0}^k |v_i v_{i+1}| \leq c_1 c_2 |st|$. \square

5.7 CONCLUSION

We have studied the problem of constructing self-approaching and increasing-chord drawings of 3-connected planar graphs and triangulations in the Euclidean and hyperbolic plane. Due to the fact that every such graph has a spanning binary cactus, and in the case of a triangulation even one that has a special type of triangulation (downward-triangulation), self-approaching and increasing-chord drawings of binary cactuses played an important role.

We showed that, in the Euclidean plane, downward-triangulated binary cactuses admit planar increasing-chord drawings, and that the condition of being downward-triangulated is essential as there exist binary cactuses that do not ad-

mit a (not necessarily planar) self-approaching drawing. Naturally, these results imply the existence of non-planar increasing-chord drawings of triangulations.

We proved that strongly monotone (and, thus, increasing-chord) drawings of trees and binary cactuses as well as greedy drawings of binary cactuses require exponential resolution in the worst case.

For planar 3-trees, which are special triangulations, we introduced α -Schnyder drawings, which have increasing chords for $\alpha \leq 30^\circ$, to show the existence of planar increasing-chord drawings.

We studied drawings in the hyperbolic plane. Here we gave a complete characterization of the trees that admit an increasing-chord drawing (which then is planar) and used it to show the existence of non-planar increasing-chord drawings of 3-connected planar graphs. For binary cactuses even a planar increasing-chord drawing exists.

Additionally, we used generalized self-approaching curves to show that Euclidean greedy drawings of trees and cactuses have bounded dilation.

Open questions

It remains open whether every 3-connected planar graph has a self-approaching or increasing-chord drawing. If this is the case, according to our example in Theorem 5.5, the construction must be significantly different from the two early proofs [AFG10; LM10] of the weak Papadimitriou-Ratajczak conjecture [PR05] (you cannot just take an arbitrary spanning binary cactus) and would prove a stronger statement.

Another question whether the method we used to construct increasing-chord drawings of planar 3-trees works for further classes of triangulations. Which triangulations admit α -Schnyder drawings for $\alpha = 30^\circ$?

It is worth noting that all self-approaching drawings we constructed are actually increasing-chord drawings. Is there a class of graphs that admits a self-approaching drawing but no increasing-chord drawing?

Do all greedy drawings have dilation bounded by some constant? If not, is this the case for all *plane* greedy drawings?

Part II

GREEDY ROUTING IN CONTINUOUS DOMAINS

PARTITIONING GRAPH DRAWINGS AND TRIANGULATED
SIMPLE POLYGONS INTO GREEDILY ROUTABLE REGIONS

A greedily routable region (GRR) is a closed subset of \mathbb{R}^2 , in which any destination point can be reached from any starting point by always moving in the direction with maximum reduction of the distance to the destination in each point of the path.

Tan and Kermarrec [TK12] proposed a geographic routing algorithm for dense wireless sensor networks based on decomposing the network area into a small number of interior-disjoint GRRs. They showed that minimum decomposition is NP-hard for polygonal regions with holes.

In this chapter, we consider minimum GRR decomposition for plane straight-line drawings of graphs. Here, GRRs coincide with self-approaching drawings of trees, a drawing style which has become a popular research topic in graph drawing. We studied this type of graph drawings in Chapter 5. We show that minimum decomposition is still NP-hard for graphs with cycles and even for trees, but can be solved optimally for trees in polynomial time, if we allow only certain types of GRR contacts. Additionally, we give a 2-approximation for simple polygons, if a given triangulation has to be respected.

This chapter is based on joint work with Martin Nöllenburg and Ignaz Rutter [NPR15; NPR17].

6.1 INTRODUCTION

In Chapter 2, we gave an overview of geographic routing, a concept of using geographic coordinates of sensor nodes for routing messages in a wireless sensor network. Routing protocols that use this notion often make use of greedy routing. Recall that greedy routing alone does not guarantee delivery, since the message might get stuck in a local minimum. We listed several approaches for overcoming such local minima, one of which is to partition the network into components, such that greedy routing inside a single component performs well; see Section 2.4.2.

One such network decomposition approach has been proposed by Tan and Kermarrec [TK12]. The authors argue that the network boundary and large holes in the network are the main source of local minima to focus on. With this reasoning, they model the network as a polygonal region with obstacles or holes inside it

*continuous domain
assumption*

and consider greedy routing inside this continuous domain; see Section 2.4. Local minima now only appear on the boundaries of the polygonal region. In this chapter, we use the same model.

Tan and Kermarrec [TK12] try to partition this region into a minimum number of polygons, in which greedy routing works between any pair of points. They call such components *greedily routable regions (GRRs)*.

For routing in the underlying network of sensor nodes corresponding to discrete points inside the polygonal region, greedy routing is used if the source and the destination nodes are in the same component, and existing techniques are used to overcome local minima. For inter-component routing, each node stores a neighbor on a shortest path to each component. This path is used to get to the component of the destination, and then intra-component routing is used. In this way, inter-component routing requires nodes to have *non-local state*, i.e., they need to store information other than that about their direct neighbors.

The number of network components in a decomposition directly reflects the number of non-local routing states of a node and determines the size of that node's routing table. Since sensor nodes typically have little available memory, it is important for them to store as small routing tables as possible. Therefore, the goal is to partition the network into a minimum number of GRRs. In this chapter, we focus on the problem of partitioning a polygonal region or a graph drawing (for which we extend the notion of a GRR) into a minimum number of GRRs. For a detailed description of an actual routing protocol based on GRR decompositions, see the original work of Tan and Kermarrec [TK12].

The authors prove that partitioning a polygon with holes into a minimum number of regions is NP-hard, and they propose a simple heuristic. Its solution may strongly deviate from the optimum even for very simple polygons; see Figures 50a and 51.

Some benchmark instances from the work of Tan and Kermarrec are networks of sensor nodes distributed on roads of a city; see Figure 12 in Section 2.4. The resulting polygonal regions are very narrow and strongly resemble plane straight-line graph drawings. Therefore, considering plane straight-line graph drawings in addition to polygonal regions is a natural adjustment of the minimum GRR partition problem.

In this chapter, we approach the problem of finding minimum or approximately minimum GRR decompositions by first considering the special case of partitioning drawings of graphs, which can be interpreted as very thin polygonal regions. We notice that in this scenario, GRRs coincide with increasing-chord drawings of trees as studied by Alamdari et al. [Ala+13]. Increasing-chord graph drawings have been considered in Chapter 5.

6.1.1 Contribution

1. First, we show that partitioning a plane graph drawing into a minimum number of increasing-chord components is NP-hard. This extends the result of Tan and

non-local state:
information about a
node that is not a
direct neighbor

Kermarrec [TK12] for polygonal regions with holes to plane straight-line graph drawings.

2. Next, we consider plane drawings of trees. We show that the problem remains NP-hard even for trees, if arbitrary types of GRR contacts are allowed. For a restriction on the types of GRR contacts, we show how to model the decomposition problem using MINIMUM MULTICUT, which provides a polynomial-time 2-approximation. We then solve the partitioning problem for trees and restricted GRR contacts optimally in polynomial time using dynamic programming.

3. Finally, we use the insights gained for decomposing graphs and apply them to the problem of minimally decomposing simple triangulated polygons into GRRs. We provide a polynomial-time 2-approximation for decompositions that are formed along chords of the triangulation.

6.2 PRELIMINARIES

In the following, let \mathcal{P} be a polygonal region, and let $\partial\mathcal{P}$ denote its boundary. For $p \in \mathcal{P}$, let $V(p)$ denote the *visibility region* of p , i.e., the set of points $q \in \mathcal{P}$ such that the line segment pq lies inside \mathcal{P} .

boundary $\partial\mathcal{P}$
visibility
region $V(p)$

Definition 6.1. For an s - t path ρ and a point $p \neq t$ on ρ , we define the forward tangent on ρ in p as the direction $\vec{d} = \lim_{\varepsilon \rightarrow 0} \{\vec{pq} \mid q \text{ succeeds } p \text{ on } \rho, \text{ and } |pq| = \varepsilon\}$.

forward tangent

Next, we formally define paths resulting from greedy routing inside \mathcal{P} . We call such paths *greedy*. Note that this definition of greediness is different from the one used in the context of greedy embeddings of graphs [PR05] that we have considered in Chapter 4.

Definition 6.2. For points $s, t \in \mathcal{P}$, an s - t -path ρ is greedy if the distance to t strictly decreases along ρ and if for every point $s' \neq t$ on ρ , the forward tangent \vec{d} on ρ in s' has the minimum angle with $\vec{s't}$ among all vectors $\vec{s'q}$ for any $q \in V(s') \setminus \{s'\}$.

greedy path

A greedy path is shown in Figure 49a. Note that such paths are polylines. The way greedy paths are defined resembles compass routing [KSU99].

6.2.1 Greedily Routable Regions

Greedily Routable Regions were introduced by Tan and Kermarrec [TK12] as follows.

Definition 6.3 ([TK12]). A polygonal region \mathcal{P} is a greedily routable region (GRR), if for any two points $s, t \in \mathcal{P}$, $s \neq t$, point s can always move along a straight-line segment within \mathcal{P} to some point s' such that $|s't| < |st|$.

GRR

Next we show that \mathcal{P} is a GRR if and only if every pair of points in \mathcal{P} is connected by a greedy path. Therefore, Definition 6.3 is equivalent to the one used in the introduction. Consider Procedure 3. We show that it produces a greedy path inside a GRR.

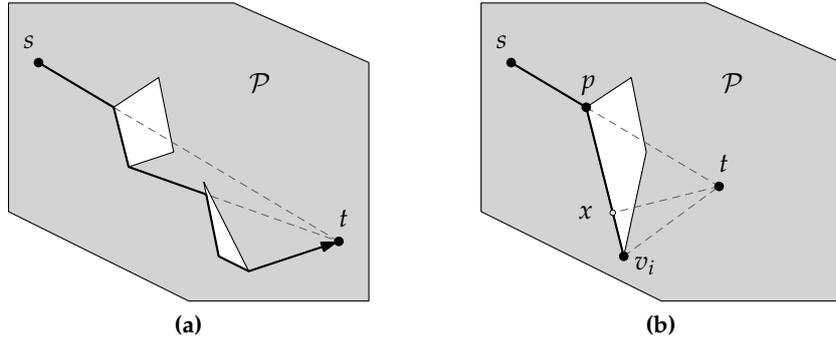


Figure 49: (a) The thick s - t -path inside the polygonal region \mathcal{P} (grey) is greedy. (b) If t is not visible, a greedy path must trace an edge until the endpoint. If it is not possible, a local minimum must exist.

<p>Procedure 3: Constructing a greedy s-t-path inside a GRR.</p> <ol style="list-style-type: none"> 1 Set $p = s$. 2 If t is visible from p, move p to t and finish the procedure. 3 Move p to the first intersection of pt and $\partial\mathcal{P}$. (Note that p itself may be the first intersection.) 4 If p is in the interior of a boundary edge v_1v_2, consider the angle between \vec{pv}_i and \vec{pt}, $i = 1, 2$. Let v_i be the vertex minimizing $\angle(\vec{pv}_i, \vec{pt})$, $i = 1, 2$ (break ties arbitrarily). If v_i is the closest point to t on the segment pv_i, move p to v_i and return to Step 2, otherwise, return failure. 5 If p coincides with the vertex v_2 incident to boundary edges v_1v_2 and v_2v_3, consider the angle between \vec{pv}_i and \vec{pt}, $i = 1, 3$. Let v_i be the vertex minimizing $\angle(\vec{pv}_i, \vec{pt})$, $i = 1, 3$ (break ties arbitrarily). Again, if v_i is the closest point to t on the segment pv_i, move p to v_i and return to Step 2, otherwise, return failure.
--

Lemma 6.1. A polygonal region \mathcal{P} is a GRR if and only if for every $s, t \in \mathcal{P}$ there exists a greedy s - t -path $\rho \subseteq \mathcal{P}$. Procedure 3 produces such a greedy path.

Proof. First, consider $s, t \in \mathcal{P}$ connected by a greedy s - t -path ρ . Then s, t satisfy the condition in Definition 6.3 using the endpoint s' of the first segment ss' of ρ .

Conversely, let \mathcal{P} be a GRR. Let s, t be two distinct points in \mathcal{P} , and consider a path ρ constructed by moving a point p from s to t according to Procedure 3. We consider the segments of ρ iteratively and show that each of them would be taken by a greedy path. Since \mathcal{P} is a GRR, every point $p \in \mathcal{P}$ can get closer to t by a linear movement. If all points on ray (p, t) sufficiently close to p are in \mathcal{P} , a greedy path would move along ray (p, t) , until it hits $\partial\mathcal{P}$. This shows that Step 3 of the procedure traces a greedy path.

Assume all points on ray (p, t) sufficiently close to p are not in \mathcal{P} . Then, p is on $\partial\mathcal{P}$. Let \vec{d}_1 and \vec{d}_2 be the two tangents in p to the paths that start at p and go

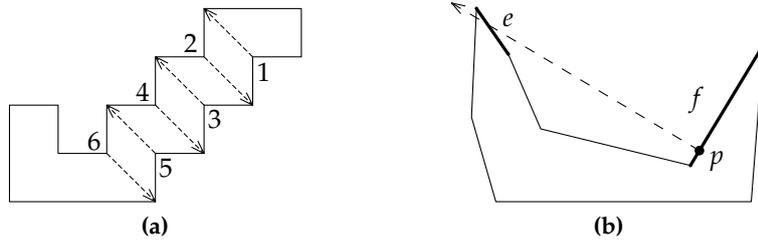


Figure 50: (a) The heuristic in [TK12] splits a non-greedy region by a bisector at a maximum inner reflex angle. If the splits are chosen in order of their index, seven regions are created, although two is minimum (split only at 6). (b) Normal ray $\text{ray}_f(p)$ and a pair of conflicting edges e, f .

along $\partial\mathcal{P}$. Let Λ be the cone of directions spanned by \vec{d}_1 and \vec{d}_2 , such that $\vec{pt} \notin \Lambda$. Then, Λ contains the directions of all possible straight-line movements from p . By Definition 6.3, for some direction $\vec{d} \in \Lambda$, we have $\angle(\vec{pt}, \vec{d}) < 90^\circ$. But then, $\min_{i=1,2} \angle(\vec{pt}, \vec{d}_i) \leq \angle(\vec{pt}, \vec{d}) < 90^\circ$. Therefore, for some $i \in \{1, 2\}$, a greedy path would continue in the direction \vec{d}_i , as does ρ . Let v_i be the endpoint of the edge containing p , such that $\vec{pv}_i = \vec{d}_i$. Therefore, $\angle tpv_i < 90^\circ$. We must show that a greedy path is traced if p follows \vec{d}_i until v_i . We have $\angle pv_it \geq 90^\circ$. Otherwise, the projection point x of t on the line through pv_i lies in the interior of the segment pv_i and is a local minimum with respect to the distance to t , which is not possible in a GRR; see Figure 49b. Therefore, when p moves in the direction \vec{d}_i towards v_i , its distance to t decreases continuously, and the forward tangent always has the minimum possible angle with respect to the direction towards t . This shows that Steps 4 and 5 of the procedure trace a greedy path and never return failure.

It follows that, when moving along ρ , point p either moves directly to t or slides along a boundary edge until it reaches one of the endpoints. Therefore, point p never reenters an edge and must finally reach t . The forward tangent on ρ always satisfies the condition of Definition 6.2, therefore, ρ is a greedy s - t -path. \square

It is easy to see that GRRs have no holes [TK12], otherwise, every hole boundary would contain a local minimum.

A *decomposition* of a polygonal region \mathcal{P} is a partition of \mathcal{P} into polygonal regions \mathcal{P}_i with no holes, $i = 1, \dots, k$, such that $\bigcup_{i=1}^k \mathcal{P}_i = \mathcal{P}$ and no $\mathcal{P}_i, \mathcal{P}_j$ with $i \neq j$ share an interior point. A decomposition of \mathcal{P} is a *GRR decomposition* if every component \mathcal{P}_i is a GRR. We shall use the terms *GRR decomposition* and *GRR partition* interchangeably. Using the concept of a *conflict relationship* between edges of a polygonal region (see Figure 50b), Tan and Kermarrec give a convenient characterization of GRRs.

GRR partition

Definition 6.4 (Normal ray). *Let \mathcal{P} be a polygonal region, $e = uv$ a boundary edge and p an interior point of uv . Let $\text{ray}_{uv}(p)$ denote the ray with origin in p orthogonal to uv , such that all points on this ray sufficiently close to p are not in the interior of \mathcal{P} .*

normal ray
 $\text{ray}_{uv}(p)$

We restate the definition of conflicting edges from [TK12].

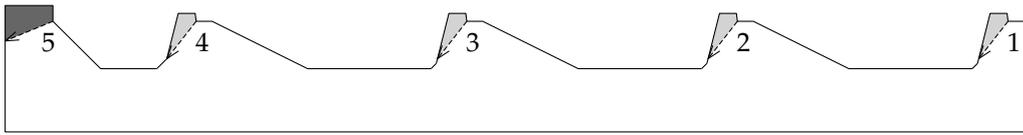


Figure 51: As an optional optimization, Tan and Kermarrec [TK12] propose to iteratively merge adjacent components, if their union is a GRR. However, this can still produce a partition whose size is worse than the optimum by a factor of $\Omega(n)$. In the shown example, the non-greedy region has been split iteratively by bisectors at maximum inner reflex angles in the shown order. If the white and the dark gray regions are merged first, all light gray regions form separate components. Choosing split 5 first partitions the polygon optimally in two GRRs.

conflicting edges

Definition 6.5 (Conflicting edges of a polygonal region). *Let e and f be two edges of a polygonal region \mathcal{P} . If for some point p in the interior of e , $\text{ray}_e(p)$ intersects f , then e conflicts with f .*

Fact 6.1 (Theorem 1 in [TK12]). *A polygonal region is a GRR if and only if it has no pair of conflicting edges.*

Now consider a plane straight-line drawing Γ of a graph $G = (V, E)$. We identify the edges of G with the corresponding line segments of Γ and the vertices of G with the corresponding points. Plane straight-line drawings can be considered as infinitely thin polygonal regions. The routing happens along the edges of Γ , and we define GRRs for graph drawings as follows.

GRRs for graph drawings

Definition 6.6 (GRRs for plane straight-line drawings). *A plane straight-line graph drawing Γ is a GRR if for any two points $s \neq t$ on Γ there exists a point s' on an edge that also contains s , such that $|s't| < |st|$.*

normal $n_e(p)$

Note that for an interior point p of an edge e of Γ there exist two normal rays at p with opposite directions. Let $n_e(p)$ denote the normal line to e at p . We define conflicting edges of Γ as follows.

Definition 6.7 (Conflicting edges of a plane straight-line drawing). *Let e and f be two edges of a plane straight-line drawing Γ . If for some point p in the interior of e , $n_e(p)$ intersects f , then e conflicts with f .*

Assume $n_e(s)$ for an interior point s on an edge e of Γ crosses another edge f in point t . Then, any movement along e starting from s increases the distance to t . We call such edges *conflicting*. It is easy to see that Γ is a GRR if it contains no pair of conflicting edges. Obviously, such a drawing Γ contains no cycles. In fact, a straight-line drawing of a tree is increasing-chord if and only if it has no conflicting edges [Ala+13], which implies the following lemma.

Lemma 6.2. *The following two properties are equivalent for a straight-line drawing Γ to be a GRR.*

1. Γ is connected and has no conflicting edges.



Figure 52: Splitting at non-vertices results in a smaller partition. (a) No pair of the thick red edges can be in the same GRR. Therefore, if no edge splits are allowed, every GRR partition has size at least 3. (b) Splitting the longest edge results in a GRR partition of size 2.

2. Γ is an increasing-chord drawing of a tree.

Since every individual edge in a straight-line drawing is a GRR, the following observation can be made on the worst-case size of a minimum GRR partition.

Observation 6.1. *A plane straight-line drawing Γ of graph $G = (V, E)$, $|E| = m$, has a GRR decomposition of size m .*

Therefore, if G is a tree, the drawing Γ has a GRR partition of size $n - 1$ for $n = |V|$.

6.2.2 Splitting graph drawings at non-vertices

Note that in a GRR partition of a plane straight-line drawing Γ of a graph $G = (V, E)$, an edge $e \in E$ does not necessarily lie in *one* GRR. Pieces of the same edge can be part of different GRRs. Allowing splitting edges at intermediate points might result in smaller GRR partitions; see Figure 52. In this section, we discuss splitting Γ at non-vertices. We will show that there are only a discrete set of $O(n^2)$ points where we might need to split edges.

Definition 6.8 (Subdivided drawing Γ_s). *Let Γ_s be the drawing created by subdividing edges of Γ as follows. For every pair of original edges $u_1u_2, u_3u_4 \in E$, let ℓ_i be the normal to u_1u_2 at u_i , $i = 1, 2$. If ℓ_i intersects u_3u_4 , we subdivide u_3u_4 at the intersection.*

subdivided
drawing Γ_s

Since we consider only the original edges of Γ , the subdivision Γ_s has $O(n^2)$ vertices.

Lemma 6.3. *Any GRR decomposition of Γ with potential edge splits can be transformed into a GRR decomposition of Γ_s in which no edge of Γ_s is split, such that the size of the decomposition does not increase.*

Proof. Consider edge uv of the subdivision Γ_s , a point x in its interior and assume an increasing-chord component C (green in Figure 53) contains vx , but not ux . We claim that we can reassign ux to C . Note that iterative application of this claim implies the lemma.

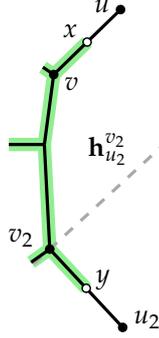


Figure 53: Proof of Lemma 6.3. Segment ux can be added to the thick green GRR C , such that the entire edge uv of Γ_s is in one GRR.

For points $p, q \in \mathbb{R}^2, p \neq q$, let \mathbf{h}_p^q denote the half-plane not containing p bounded by the line through q orthogonal to the segment pq . Recall from Chapter 5 that if segment pq is on the path from vertex p to vertex r in an increasing-chord tree drawing then $r \in \mathbf{h}_p^q$ [Ala+13].

Let u_2v_2 be an original edge of Γ such that v_2 is in C , as well as a subsegment yv_2 of u_2v_2 with a non-zero length containing v_2 . Since segment yv_2 is on the y - v -path in C , the half-plane $\mathbf{h}_{u_2}^{v_2} = \mathbf{h}_y^{v_2}$ contains v , and its boundary does not cross uv by the construction of Γ_s . Thus, $\mathbf{h}_{u_2}^{v_2}$ contains uv . In this way, we have shown that no normal to an edge of C crosses uv .

Furthermore, $\mathbf{h}_u^v = \mathbf{h}_x^v$. Since $C - xv$ lies entirely in $\mathbf{h}_x^v = \mathbf{h}_u^v$, this shows that no normal of uv crosses another edge of C . It follows that the union of C and ux contains no conflicting edges and, therefore, is increasing-chord by Lemma 6.2.

Finally, removing ux from a component C' containing it does not disconnect C' , since no edge or edge part is attached to x (or an interior point of ux). Since $C' - ux$ is connected and C' is a GRR, $C' - ux$ is also a GRR. \square

6.2.3 Types of GRR contacts in plane straight-line graph drawings

We distinguish the types of contacts that two GRRs can have in a GRR partition of a plane straight-line graph drawing.

Definition 6.9 (Proper, non-crossing and crossing contacts). *Consider two drawings Γ_1, Γ_2 of trees with the only common point p .*

1. Γ_1 and Γ_2 have a proper contact if p is a leaf in at least one of them.
2. Γ_1 and Γ_2 have a non-crossing contact if in the clockwise ordering of edges of Γ_1 and Γ_2 incident to p , all edges of Γ_1 (and, thus, also of Γ_2) appear consecutively.
3. Γ_1 and Γ_2 are crossing or have a crossing contact if in the clockwise ordering of edges of Γ_1 and Γ_2 incident to p , edges of Γ_1 (and, thus, also of Γ_2) appear non-consecutively.

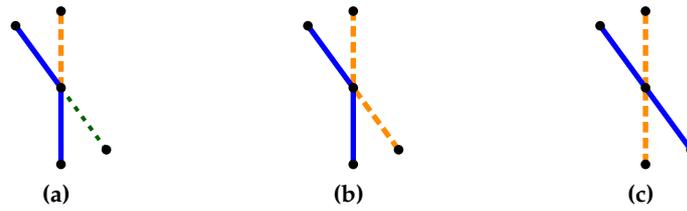


Figure 54: (a) Proper GRR contact; (b) non-crossing contact which is not proper and (c) crossing contact.

The first part of Definition 6.9 allows GRRs to only have contacts as shown in Figure 54a and forbids contacts as shown in Figures 54b, 54c. The second part allows contacts as those in Figure 54b, but forbids the contacts in Figure 54c.

Note that a contact of two trees Γ_1, Γ_2 with a single common point p is either crossing or non-crossing. Moreover, if the contact of Γ_1 and Γ_2 is proper, then it is necessarily non-crossing, since for a proper contact, Γ_1 or Γ_2 has only one edge incident to p , therefore, all edges of Γ_1 and of Γ_2 appear consecutively around p .

We shall show that for trees, restricting ourselves to GRR decompositions with only non-crossing contacts makes the otherwise NP-complete problem of finding a minimum GRR partition solvable in polynomial time.

6.3 NP-COMPLETENESS FOR GRAPHS WITH CYCLES

We show that finding a minimum decomposition of a plane straight-line drawing Γ into increasing-chord trees is NP-hard. This extends the NP-hardness result by Tan and Kermarrec [TK12] for minimum GRR decompositions of polygonal regions with holes to plane straight-line drawings.

Note that in the graph drawings used for our proof, all GRRs will have *proper contacts*; see Definition 6.9. Moreover, the graph drawings can be turned into thin polygonal regions in a natural way by making them slightly “thicker”, and the proof can be reused as another proof for the NP-hardness result in [TK12].

Both our NP-hardness proof and the proof in [TK12] are reductions from the NP-complete problem PLANAR 3SAT [Lic82]. Recall that a Boolean 3SAT formula φ is called *planar*, if the corresponding variable-clause graph G_φ having a vertex for each variable and for each clause and an edge for each occurrence of a variable (or its negation) in a clause is a planar graph. In fact, G_φ can be drawn in the plane such that all variable vertices are aligned on a vertical line and all clause vertices lie either to the left or to the right of this line and connect to the variables via E- or \exists -shapes [KR92]; see Figure 55.

The basic idea of the gadget proof is as follows. Using a number of building blocks, or *gadgets*, we construct a plane straight-line drawing Γ_φ , whose geometry mimics the variable-clause graph G_φ drawn as described above. We construct Γ_φ in a way such that its minimum GRR decompositions are in correspondence with the truth assignments of the PLANAR 3SAT formula φ .

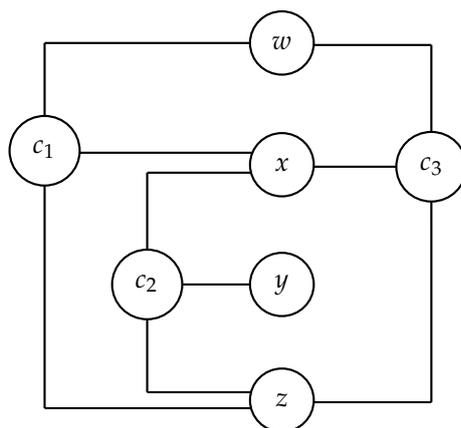


Figure 55: An orthogonal graph drawing of the variable-clause graph G_ϕ for a planar 3SAT formula $\phi = (w \vee x \vee z) \wedge (\bar{x} \vee y \vee \bar{z}) \wedge (\bar{w} \vee \bar{x} \vee \bar{z})$.

variable gadgets

The variable gadgets in [TK12] are cycles formed by T-shaped polygons which can be made arbitrarily thin. Thus, in the case of plane straight-line drawings we can use very similar variable gadgets (see Figure 56). The clause gadgets in [TK12], however, are squares, at which three variable cycles meet. This construction cannot be adapted for straight-line plane drawings, and we have to construct a significantly different clause gadget; see Figure 58.

We define a variable gadget as a cycle of alternating vertical and horizontal segments. The tip of each segment touches an interior point of the next segment. We can join pairs of consecutive segments into a GRR by assigning each vertical segment either to the next or to the previous horizontal segment on the cycle. In this way, the variable loop is partitioned either in \top -shapes and \perp -shapes or in \neg -shapes and \vdash -shapes; see Figure 56.

black and white points

Consider a variable gadget consisting of k T-shapes as shown in Figure 56. On each T-shape we place one black and one white point as shown in the figure. The points are placed in such a way that neither two black points nor two white points can be in one increasing-chord component. Thus, a minimum GRR decomposition of a variable gadget contains at least k components. If it contains exactly k components, then each component must contain one black and one white point, and there are exactly two possibilities. Each black point has exactly two white points it can share a GRR with, and once one pairing is picked, it fixes all the remaining pairings. The corresponding possibilities are shown in Figure 56a and 56b and will be used to encode the values *true* and *false*, respectively. For the pairing of the black and white points corresponding to the true state, the variable loop can be partitioned in \top -shapes and \perp -shapes, and for the pairing corresponding to the false state, it can be partitioned in \neg -shapes and \vdash -shapes.

arm gadgets

To pass the truth assignment of a variable to a clause it is part of, we use *arm gadgets*. Arm gadgets are extensions of the variable gadget. To add an arm gadget to the variable, we substitute several \top - or \perp -shapes from the variable loop by a more complicated structure. Figure 56c shows such extensions for all arm

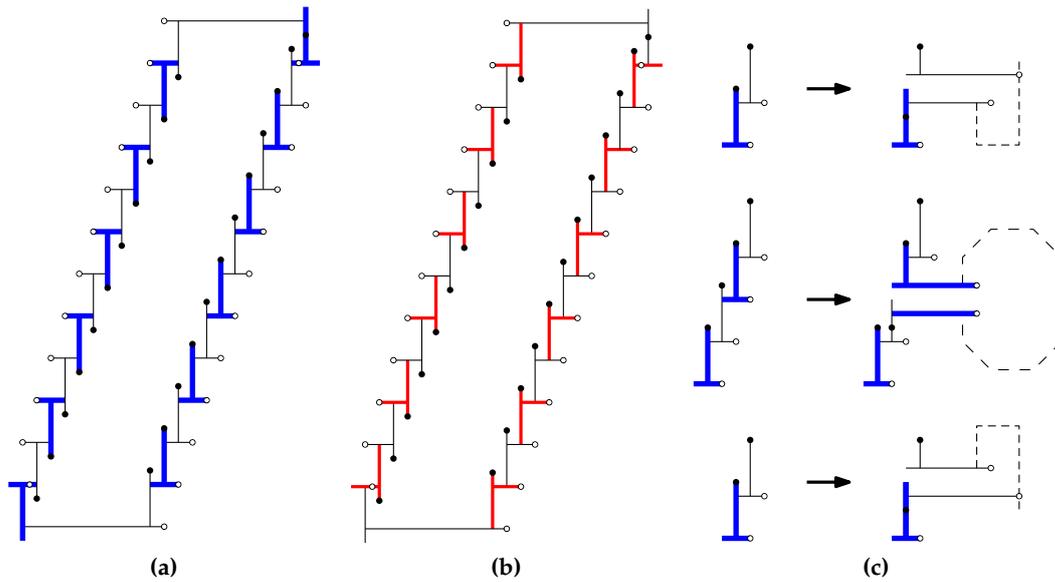


Figure 56: Variable gadget and the two possibilities to pair vertical and horizontal segments to make GRRs: (a) *true* variable state: T-shapes and L-shapes; (b) *false* variable state: inverted T-shapes and inverted L-shapes. (c) Extending the variable gadgets to create the upper, middle and lower arm gadgets by substituting T-shapes of the variable gadget.

types pointing to the right, the other case is symmetric. In this way, for a variable, we can create as many arms as necessary. Each variable loop will have one arm extension for each occurrence of the corresponding variable in a clause in φ . The working principle for the arm gadgets is the same as for the variable gadgets. The drawing created by the variable cycle and the arm extensions (the *variable-arm loop*) will once again contain distinguished black and white points, such that only one black and one white point can be in a GRR. However, for variable-arm loops, the cycles formed by segments of varying orientation are more complicated than the loop in Figure 56. For example, for some arm types we use segments of slopes ± 1 in addition to vertical and horizontal segments.

variable-arm loop

In total twelve variations of the arm gadget will be used, depending on the position of the literal in the clause, the position of the clause, and whether the literal is negated or not. Since in G_φ each clause c connects to three variables, we denote these variables or literals as the *upper*, *middle*, and *lower* variables of c depending on the order of the three edges incident to c in the one-bend orthogonal drawing of G_φ used by Knuth and Raghunathan [KR92]; see Figure 55. Similarly, an arm of c is called an *upper*, *middle*, or *lower* arm if it belongs to a literal of the same type in c . An arm is called a *right* (resp. *left*) arm if it belongs to a clause that lies to the right (resp. to the left) of the vertical variable line. Finally, an arm of c is *positive* if the corresponding literal is positive in c and it is *negative* otherwise.

upper, middle, lower arms

left and right, positive and negative arms

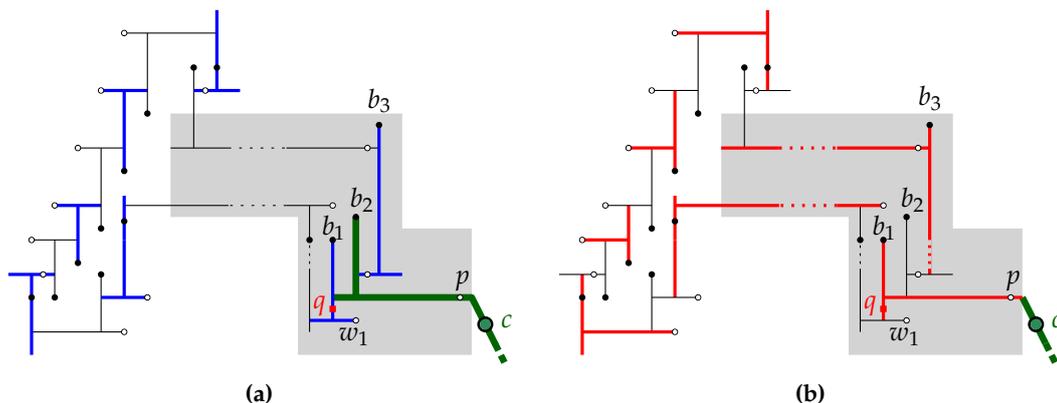


Figure 57: Variable gadget with a right upper positive arm (shaded region). (a) *true* and (b) *false* states.

The basic principle of operation of any arm gadget is the same; as an example consider the right upper positive arm in Figure 57. Figures 60, 61, 62 and the proof of Property 2 cover the remaining arm types.

The positive and the negative arms are differentiated by an additional structure that switches the pairing of the black and white points close to the part of the arm that touches the clause gadget; for example, compare Figures 57b and 62a. By this inversion, for a fixed truth assignment of the variable, the \top - and \perp -shapes next to the clause are turned into \vdash - and \dashv -shapes, and vice versa. In this way, the inverted truth assignment of the corresponding variable is passed to the clause.

Note that each arm can be arbitrarily extended both horizontally and vertically to reach the required point of its clause gadget. We select again black and white points (also called *distinguished* points) on the line segments of the arm gadget.

clause gadget

The *clause gadget* (the thickest green polyline in Figure 58, partly drawn in Figure 57) is a polyline which consists of six segments. The first segment has slope 2, the second is vertical, the third has slope -1 , the fourth has slope 1, the fifth is vertical, and the sixth has slope -2 . Each clause gadget connects to the long horizontal segments of the arms of three variable gadgets. The three connecting points of the clause gadget are the start and end of the polyline as well as its center, which is the common point of the two segments with slopes ± 1 .

We shall prove the following property which is crucial for our construction.

Property 1. 1. Consider a drawing Γ_i of a variable gadget together with all of its arms. Then, neither two black nor two white points on Γ_i can be in one GRR. In a minimum GRR decomposition of Γ_i , each component has one black and one white point, and exactly two such pairings of points are possible, one for each truth assignment.

2. Consider two such drawings Γ_i, Γ_j for two different variables. Then, no distinguished point of Γ_i can be in the same GRR as a distinguished point of Γ_j .

Proof. Part (1) of Property 1 extends the same property that we already showed for variable gadgets without arms to the case including all arms. It is an imme-

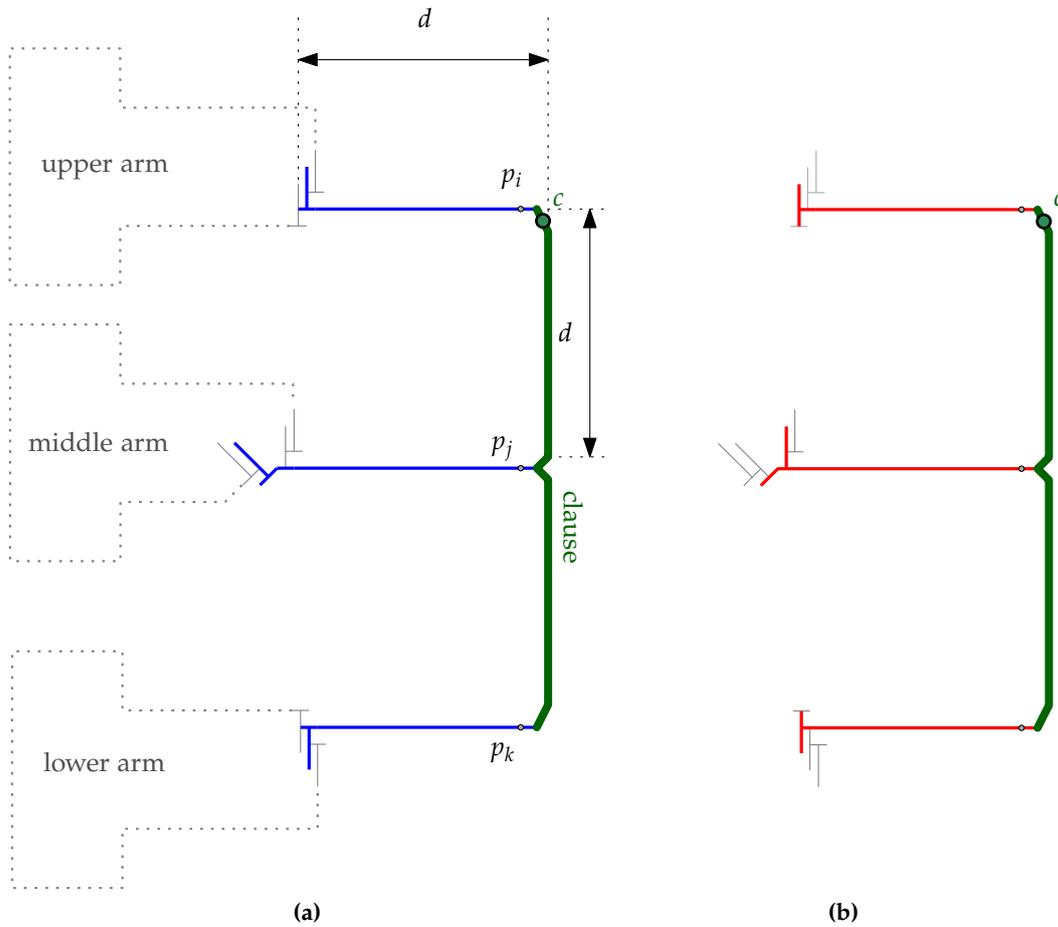


Figure 58: Clause gadget (thick green). (a) *true* and (b) *false* state of the involved literals.

diate consequence of the way we constructed the arm gadgets and placed the distinguished points; see Figures 57, 60, 61, 62.

Part (2) follows from the way the arms are connected by a clause, i.e., in Figure 58 no pair of points from p_i, p_j, p_k can be in the same GRR, since the three points lie on three horizontal segments and are vertically collinear. \square

The clause gadget is connected to an arm by a horizontal segment with a distinguished point p on its end, which is either black or white depending on the arm type. Each clause has one special point c chosen as shown in Figure 58.

We show that c and p can be in the same GRR in a minimum GRR decomposition if and only if the variable gadget containing p is in the state that satisfies the clause.

Property 2. 1. *In a minimum GRR decomposition, the special point c of a clause gadget can share a GRR with a black or white point of an arm gadget if and only if the corresponding literal is in the true state.*

2. *If a variable assignment satisfies a clause, then its entire clause gadget can be contained in a GRR of an arm corresponding to a true literal.*

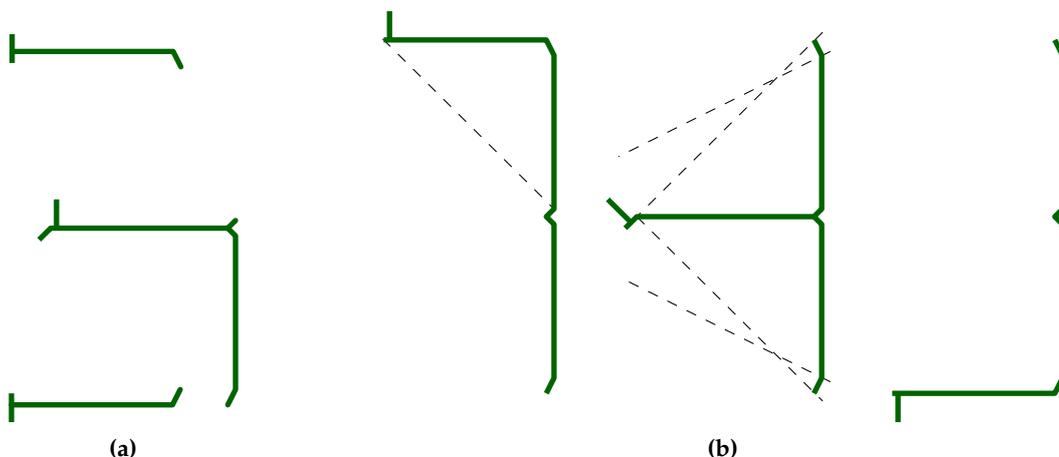


Figure 59: Merging the clause gadget with GRRs from the arm loops. (a) None of the three components is a GRR. (b) All three components are GRRs; see the dashed normals.

Proof. For each arm gadget we select a special *red* point q ; see Figure 57. Point q is neither white nor black. By Property 1, in a minimum GRR decomposition, point q must be in a GRR together with one black and one white point.

For the various arm types, if points q and p are in the same GRR, we shall show that this GRR cannot contain the entire clause gadget and, in particular, cannot contain point c . This is illustrated in Figure 59a.

Furthermore, we shall show that if the literal is in the *true* state, then points p and q are in different GRRs, and the GRR containing p can be merged with the entire clause gadget, including c . For example, in Figure 58a, each variable is in a state that satisfies the clause. The lengths of the thick segments are chosen such that each thick blue component can be merged with the clause gadget (thickest green) into a single GRR, as shown in Figure 59b.

(i) We first show the lemma for a positive right upper arm. We use the notation from Figure 57 to refer to the distinguished points. In the *true* state of the variable (see Figure 57a), points w_1, b_1 and q are in the same GRR. Points b_2 and p are in another GRR (e.g., the thickest green one in Figure 57) which can contain the distinguished point c of the clause.

In the *false* state of the variable (see Figure 57b), the points b_1 and p are in the same GRR. Moreover, point q can share a GRR with exactly one point from b_1, b_2 or b_3 . But if q were with b_2 or b_3 , then b_1 would be disconnected from any white point, a contradiction to the minimality of the decomposition. Thus, points q, b_1 and p are in the same GRR, which cannot contain a point of the clause.

(ii) We now show the lemma for a negative right lower arm. We use the notation from Figure 60. In the *false* state of the variable (which corresponds to the *true* state of the considered literal), points w_1, b_1 and q are in the same GRR; see Figure 60a. Points b_2 and p are in another GRR (e.g., the very thick green one in

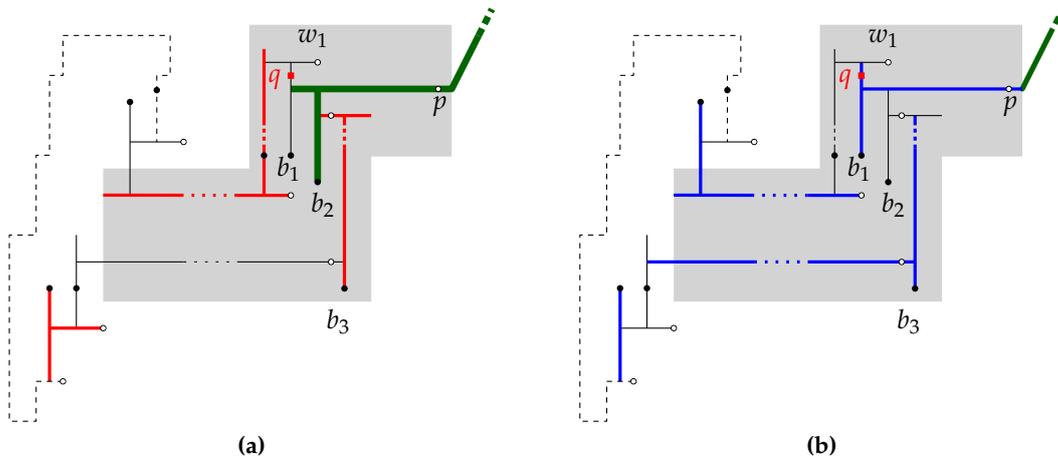


Figure 60: Right lower negative arm gadget. (a) *false* and (b) *true* variable state. Thin dashed lines indicate that the variable-arm loop continues.

Figure 57) which can contain the entire clause; see the lower arm in Figure 58 and the corresponding merged component in Figure 59b.

Now consider a *true* state of the variable; see Figure 60b. Point q shares a GRR with exactly one point from b_1, b_2 or b_3 . If q is with b_2 or b_3 , then b_1 is disconnected from any white point, a contradiction to the minimality of the decomposition. Thus, points q, b_1 and p are in the same GRR, which cannot contain a point of the clause.

(iii) Next, consider a positive right middle arm; see Figure 61. We identify points p and b_1 . Point b_1 is either with w_0 (*true* state of the variable) or w_1 (*false* state of the variable).

In the *true* state, points b_1 and w_0 are in one GRR, which cannot contain q . This GRR can be merged with the clause gadget; see Figure 61a, 58 and 59b.

In the *false* state, points b_1, w_1 and q are in one GRR, which cannot contain point c of the clause.

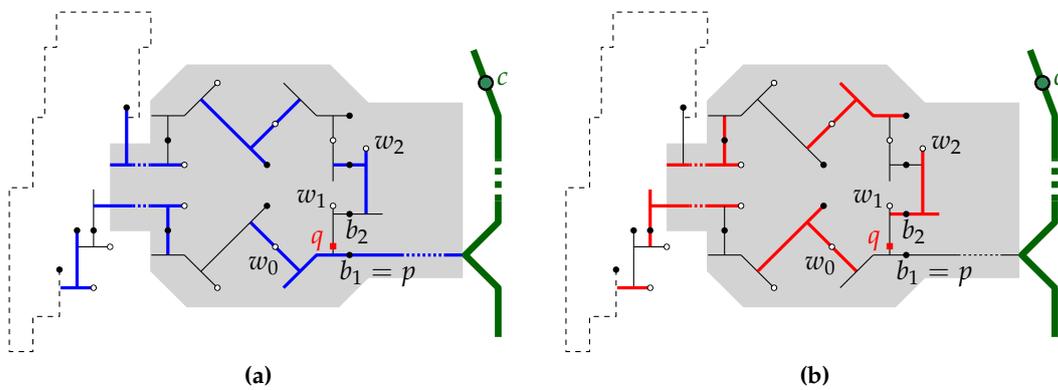


Figure 61: Right positive middle arm gadget. (a) *true* and (b) *false* variable state.

- (iv) To construct the negative right upper arm, the positive right lower arm and the negative right middle arm, we invert the arm gadgets constructed before. The inverted gadgets are shown in Figure 62. The proofs are analogous to the respective non-inverted cases.
- (v) The left arms are constructed by mirroring. □

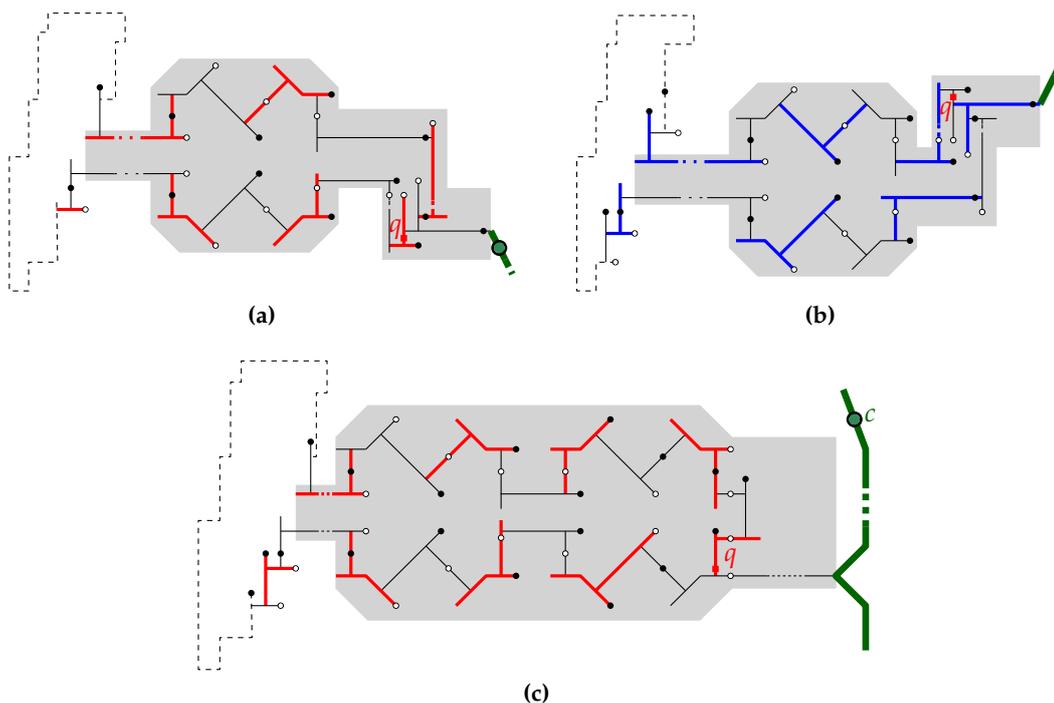


Figure 62: The remaining three right arms in the satisfying variable state. (a) negative right upper arm, (b) the positive right lower arm and (c) the negative right middle arm.

Finally, we can prove the NP-hardness result by showing that any satisfying truth assignment for a formula φ yields a GRR decomposition into a fixed number k of GRRs, where k is the total number of black points in our construction. Likewise, using Property 1 and 2, we can show that any decomposition into k GRRs necessarily satisfies each clause in φ .

Theorem 6.1. *For $k \in \mathbb{N}_0$, deciding whether a plane straight-line drawing can be partitioned into k increasing-chord components is NP-complete.*

Proof. First, we show that the problem is in NP. Given a plane straight-line drawing Γ , we construct its subdivision Γ_s as described in Section 6.2.2. By Lemma 6.3, it is sufficient to consider only partitions of edges in Γ_s into k components. To verify a positive instance, we non-deterministically guess the partition of the edges of Γ_s into k components. Testing if each component is a tree and if it is increasing-chord can be done in polynomial time.

Next, we show NP-hardness. Given a Planar 3SAT formula φ , we construct a plane straight-line drawing Γ_φ using the gadgets described above. It is easy to see that Γ_φ can be constructed on an integer grid of polynomial size and in polynomial time. Let k be the number of black points produced by the construction. Note that k is $O(m+n)$, where n is the number of variables and m the number of clauses in φ . We claim that Γ_φ can be decomposed into k GRRs if and only if φ is satisfiable.

Consider a truth assignment of the variables satisfying φ . We decompose each variable gadget and the attached arms as intended in our gadget design, which yields exactly k GRRs. By Property 2, each clause gadget can be merged with the GRR of the arm of a literal which satisfies the clause. Therefore, we have k GRRs in total.

Conversely, consider a decomposition of Γ_φ into k GRRs. Then, each variable and the attached arms must be decomposed minimally and, by Property 1, must be either in the *true* or in the *false* state. Furthermore, each special point c of a clause must be in a component belonging to one of the arms of the clause. But then, the corresponding variable must satisfy the clause by Property 2. This induces a satisfying variable assignment for φ . \square

6.4 TREES

In this section we consider *greedy tree decompositions*, or GTDs. For trees, greedy regions correspond to increasing-chord drawings. Note that increasing-chord tree drawings are either subdivisions of $K_{1,4}$, subdivisions of the *windmill* graph (three caterpillars with maximum degree 3 attached at their “tails”) or paths; see the characterization by Alamdari et al. [Ala+13].

In the following, we consider a plane straight-line drawing Γ of a tree $T = (V, E)$, with $|V| = n$. As before, we identify the tree with its drawing, the vertices with the corresponding points and the edges with the corresponding line segments. We want to partition it into a minimum number of increasing-chord subdrawings. In such a partition, each pair of components shares at most one point.

Recall that a contact of two trees Γ_1, Γ_2 with a single common point p is either crossing or non-crossing; see Definition 6.9. Also, recall that proper contacts are non-crossing. Let Π_{all} be the set of all GRR partitions of the plane straight-line tree drawing Γ . Let Π_{nc} be the set of GRR partitions of Γ , in which every pair of GRRs has a non-crossing contact. Finally, let Π_{p} be the set of GRR partitions of Γ , in which every pair of GRRs has a proper contact. It holds: $\Pi_{\text{p}} \subseteq \Pi_{\text{nc}} \subseteq \Pi_{\text{all}}$. For minimum partitions $\pi_{\text{p}}, \pi_{\text{nc}}, \pi_{\text{all}}$ from $\Pi_{\text{p}}, \Pi_{\text{nc}}, \Pi_{\text{all}}$, respectively, we have $|\pi_{\text{all}}| \leq |\pi_{\text{nc}}| \leq |\pi_{\text{p}}|$.

We show that finding a minimum GTD of a plane straight-line tree drawing is NP-hard; see Section 6.4.1. In Section 6.4.2, we show that the problem becomes polynomial if we consider GRR partitions in which GRRs have only non-crossing

contacts, i.e., partitions from Π_{nc} . The same holds if we only consider GRR partitions in which GRRs only have proper contacts, i.e., partitions from Π_p .

6.4.1 NP-completeness

We show that if GRR crossings as in Definition 6.9 are allowed, deciding whether a partition of given size exists is NP-complete.

The following problem is known as PARTITION INTO TRIANGLES (PIT). It has been shown to be NP-complete by Čustić et al. [ČKW15, Proposition 5.1] and will be useful for our hardness proof.

Problem 2 (PIT). *Given a tripartite graph $G = (V, E)$ with tripartition $V = V_1 \cup V_2 \cup V_3$, where $|V_1| = |V_2| = |V_3| = q$. Does there exist a set T of q triples in $V_1 \times V_2 \times V_3$, such that every vertex in V occurs in exactly one triple and such that every triple induces a triangle in G ?*

It is easy to show that the following, similar problem PARTITION INTO INDEPENDENT TRIPLES (PIIT) is NP-complete as well.

Problem 3 (PIIT). *Given a tripartite graph $G = (V, E)$ with tripartition $V = V_1 \cup V_2 \cup V_3$, where $|V_1| = |V_2| = |V_3| = q$. Does there exist a set T of q triples in $V_1 \times V_2 \times V_3$, such that every vertex in V occurs in exactly one triple and such that no two vertices of a triple are connected by an edge in G ?*

Lemma 6.4. *PIIT is NP-complete.*

Proof. It is easy to see that PIIT is in NP. For NP-hardness, consider a graph $G = (V, E)$ from an instance of PIT. We construct $G' = (V, E')$ with $E' = \{uv \mid uv \notin E, u \in V_i, v \in V_j, i \neq j \text{ for } i, j = 1, 2, 3\}$. In this way, a triple from $V_1 \times V_2 \times V_3$ induces a triangle in G if and only if it is independent in G' . Therefore, PIT can be reduced to PIIT in polynomial time. \square

We now show that deciding whether a GRR partition of a plane straight-line tree drawing of given size exists is NP-complete even for subdivisions of a star.

Theorem 6.2. *Given a plane straight-line drawing Γ of a tree $T = (V, E)$, which is a subdivision of a star with $3q$ leaves, it is NP-complete to decide whether Γ can be partitioned into q GRRs.*

Proof. The proof that the problem is in NP is analogous to the corresponding proof of Theorem 6.1.

To prove NP-hardness, we present a polynomial-time reduction from PIIT. Consider the tripartite graph $G = (V, E)$ with tripartition $V = V_1 \cup V_2 \cup V_3$ from an instance $\Pi = (G, V_1, V_2, V_3, q)$ of PIIT, where $|V_1| = |V_2| = |V_3| = q$. We may assume $q \geq 3$. We show how to construct a plane straight-line drawing Γ of a subdivision of a star in polynomial time, such that Γ can be partitioned into q GRRs if and only if Π is a yes-instance of PIIT. Figure 63 shows an example of such a construction for $q = 3$.

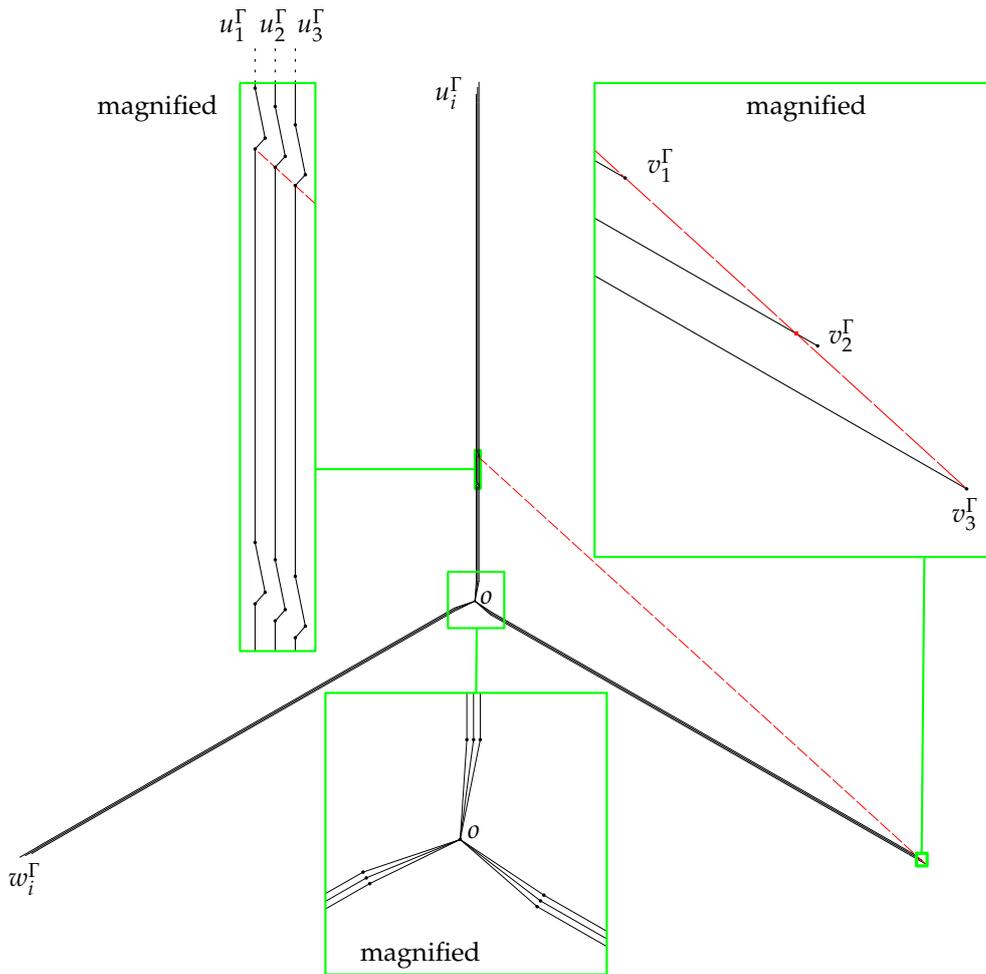


Figure 63: Reduction from a PIIT instance with $q = 3$ for the proof of Theorem 6.2.

We use the following basic ideas to construct the drawing Γ . Let o be the center of Γ . Each vertex v of G corresponds to a leaf vertex v^Γ of Γ . The leaves of Γ are partitioned into three sets corresponding to V_1, V_2, V_3 . Consider a pair of vertices $u \in V_i, v \in V_j$. If $i = j$, the angle that the u^Γ - v^Γ -path has at point o in our construction is at most 12° . Therefore, u and v can not be in the same GRR. For $i \neq j$, however, the angle that the u^Γ - v^Γ -path has at point o is between 106° and 134° . We construct the o - u^Γ and o - v^Γ paths in such a way that the u^Γ - v^Γ -path is increasing-chord if and only if edge uv is not in G .

The path from o to v^Γ takes a left turn of at most 12° and then continues as a straight line, except for at most q dents; see the left magnified part of Figure 63. Each dent is used to realize exactly one edge from G . For a pair of vertices $u \in V_i, v \in V_j, j \equiv i + 1 \pmod{3}$ with edge uv in G , the o - u^Γ -path has a dent with a normal crossing the o - v^Γ -path. Furthermore, no normal to this dent crosses the o - w^Γ -path for any vertex $w \in V_j \cup V_k \setminus \{v\}$, for $k \equiv i + 2 \pmod{3}$. Consider the example in Figure 63. Assume that there is an edge u_3v_2 in G . Then, the o - u_3^Γ -

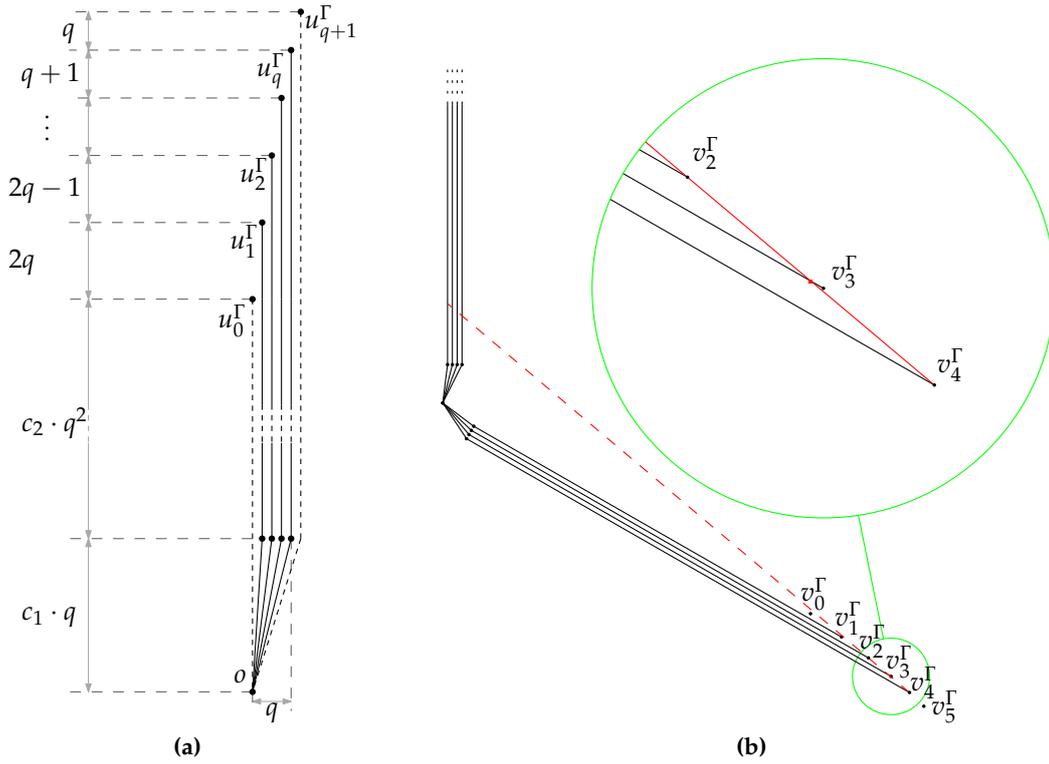


Figure 64: Constructing Γ from Π for the proof of Theorem 6.2. (a) Choosing coordinates for points $o, u_0^\Gamma, \dots, u_{q+1}^\Gamma$. (b) Determining dent positions on the $o-u_i^\Gamma$ -paths.

path has a dent whose normal (dashed red) crosses the $o-v_2^\Gamma$ -path, but not the paths from o to $v_1^\Gamma, v_3^\Gamma, w_1^\Gamma, w_2^\Gamma$ and w_3^Γ .

We now describe the procedure to construct Γ from Π in detail. We will make sure that all vertices of Γ have rational coordinates with numerators and denominators in $O(n^2)$. Let $V_1 = \{u_1, \dots, u_q\}$, $V_2 = \{v_1, \dots, v_q\}$ and $V_3 = \{w_1, \dots, w_q\}$. For the construction, we introduce dummy points $u_0^\Gamma, u_{q+1}^\Gamma, v_0^\Gamma, v_{q+1}^\Gamma, w_0^\Gamma, w_{q+1}^\Gamma$, which do not lie on Γ . For all $i = 0, \dots, q + 1$, we shall have $|ou_i^\Gamma| = |ov_i^\Gamma| = |ow_i^\Gamma|$.

We first show how to choose coordinates for points $o, u_0^\Gamma, \dots, u_{q+1}^\Gamma$; see Figure 64a. We approximate 120° rotation using the angle $\alpha \approx 120.51^\circ$ with $\cos \alpha = -\frac{33}{65}$ and $\sin \alpha = \frac{56}{65}$. The points v_i^Γ are acquired from u_i^Γ by a clockwise rotation by α at o , and the points w_i^Γ are acquired from u_i^Γ by a counterclockwise rotation by α at o . Then, $\angle u_i^\Gamma o v_i^\Gamma = \angle u_i^\Gamma o w_i^\Gamma = \alpha$ and $\angle v_i^\Gamma o w_i^\Gamma = 360^\circ - 2\alpha \approx 118.98^\circ$.

Let point o have coordinates $(0, 0)$. For $i = 1, \dots, q$, let the first segment of the $o-u_i^\Gamma$ -path have its other endpoint in $(i, c_1 q)$ for a constant c_1 . For $i = 0, \dots, i + 1$, point u_i^Γ has x -coordinate i . Let y_i denote the y -coordinate of u_i^Γ . We set $y_0 = c_1 q + c_2 q^2$ for a constant c_2 . For $i = 1, \dots, q$, we set $y_i = y_{i-1} + 2q + 1 - i$; see Figure 64a. Thus, for $i = 0, \dots, q + 1$, points u_i^Γ lie on a parabola that opens down. Note that all vertices of Γ constructed so far are integers in $O(n^2)$. We set $c_1 = 5$ and $c_2 = 40$.

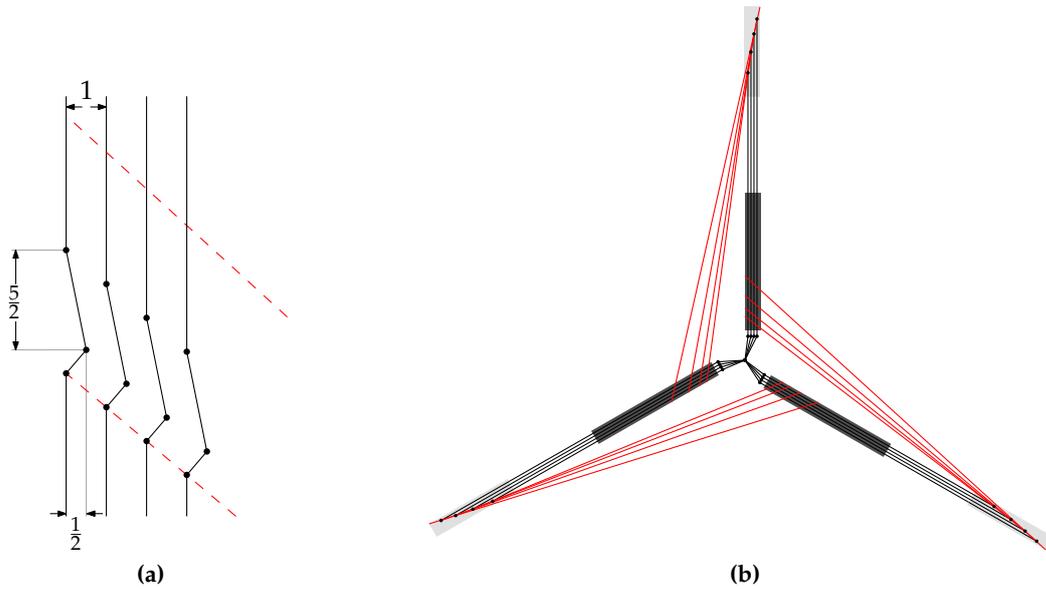


Figure 65: Proof of Theorem 6.2 continued. (a) Constructing dents on the $o-u_i^\Gamma$ -paths. (b) All dents lie inside the three dark gray regions.

Next, we show how to construct the dents on the $o-u_i^\Gamma$ -paths. For edge $u_i v_j$ in G , $i, j = 1, \dots, q$, consider the straight line through $v_{j-1}^\Gamma v_{j+1}^\Gamma$; see the dashed red line in Figure 64b for $j = 3$. Consider the intersection of this line and the vertical line through u_i^Γ . The coordinates of that intersection are rational numbers with numerators and denominators in $O(n^2)$. It is easy to show that this intersection has y -coordinates between $20q$ and $80q$.

At the intersection, we place a dent consisting of two segments; see Figure 65a. The first segment of the dent has positive slope and is orthogonal to $v_{j-1}^\Gamma v_{j+1}^\Gamma$. Its projection on the x axis has length $\frac{1}{2}$. The second segment has the negative slope of -5 . It is easy to verify that the line through $v_j^\Gamma v_{j+2}^\Gamma$ (the upper red dashed line in Figure 65a) has distance at least $\frac{c_2}{8} = 5$ from the lowest point of the dent. Therefore, the dent fits between the two dashed red lines. Note that all three vertices of the dent have coordinates that are rational numbers with numerators and denominators in $O(n^2)$.

By the choice of the slopes, no normal to either one of the two dent segments crosses ow_k^Γ for $k = 0, \dots, q + 1$. Furthermore, no normal on the second segment crosses ov_k^Γ for $k = 0, \dots, q + 1$, and a normal to the first segment only crosses ov_k^Γ for $k = j$. In this way, the dent ensures that u_i^Γ and v_j^Γ can not be in the same GRR, and it does not prohibit any other vertex pair (u_k^Γ and v_ℓ^Γ , v_k^Γ and w_ℓ^Γ , w_k^Γ and v_ℓ^Γ , for $k, \ell = 1, \dots, q$) from being in the same GRR. Finally, for each leaf vertex u_i^Γ , we add the missing segments on the vertical line through u_i^Γ to connect o and u_i^Γ by a path. Analogously, we construct the $o-v_i^\Gamma$ - and the $o-w_i^\Gamma$ -paths.

Note that by our construction, the dent normals do not cross other dents on the paths from o to the leaves from another partition; see Figure 65b, where the dents

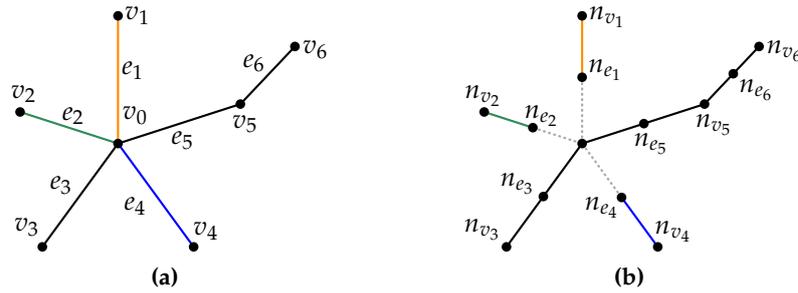


Figure 66: (a) Tree drawing decomposed in GRRs. Edge pairs $\{e_1, e_2\}, \dots, \{e_4, e_5\}, \{e_5, e_1\}$ as well as $\{e_1, e_6\}, \{e_4, e_6\}$ are conflicting. (b) MINIMUM MULTICUT instance constructed according to the proof of Proposition 6.5. No edge orientation provides directed paths between all terminal pairs. Dashed edges form a solution.

lie in the dark gray rectangles, and the crossings of dent normals and paths from o to the leaves from another partition lie in the light gray rectangles. It follows that for $i, j = 1, \dots, q$, the $o-u_i^\Gamma$ - and the $o-v_j^\Gamma$ -path can be merged into one GRR, if no dent corresponding to edge $u_i v_j$ in G exists on the $o-u_i^\Gamma$ -path in Γ .

From the construction of Γ , it follows that a pair of leaves x^Γ and y^Γ can be in the same GRR if and only if the corresponding vertices x, y are in different partitions of V and edge xy is not in G . Therefore, triples of leaves $x^\Gamma, y^\Gamma, z^\Gamma$ for which $x^\Gamma, y^\Gamma, z^\Gamma$ can be in the same GRR, are in one to one correspondence to independent triples from $V_1 \times V_2 \times V_3$ in G . Therefore, Γ can be partitioned into q GRRs if and only if Π is a yes-instance of PIIT. Note that Γ can be constructed in polynomial time and that all coordinates of vertices in Γ are rational numbers with numerators and denominators in $O(n^2)$. \square

6.4.2 Polynomial-time algorithms for restricted types of contacts

We now make a restriction by only allowing non-crossing contacts.

First, assume T is split only at its vertices. As shown in Section 6.2.2, we can drop this restriction and adapt our algorithms to compute minimum or approximately minimum GRR decompositions of plane straight-line tree drawings which allow splitting tree edges at interior points. Note that the construction in the proof of Lemma 6.3 preserves the non-crossing property of GRR contacts.

We start in Section 6.4.2.1 and use the well-known problem MINIMUM MULTICUT to compute a 2-approximation for minimum GTDs for the scenario in which GRRs are only allowed to have proper contacts. A similar approach will be used in Section 6.5 to compute minimum GRR decompositions of triangulated polygons. After that, in Section 6.4.2.2, we present an exact, but more complex approach for computing GTDs, which also allows non-crossing contacts.

6.4.2.1 2-approximation using Multicut

We show how to partition the edges of T into a minimum number of increasing-chord components with proper contacts using MINIMUM MULTICUT on trees. For a given edge-weighted graph $G = (V, E)$ and a set of terminal pairs $\{(s_1, t_1), \dots, (s_k, t_k)\}$, an edge set $S \subseteq E$ is a *multicut* if removing S from G disconnects each pair s_i, t_i , $i = 1, \dots, k$. A multicut is minimum if the total weight of its edges is minimum.

MINIMUM
MULTICUT

For the complexity of MINIMUM MULTICUT on special graph types, see the survey by Costa et al. [CLR05]. Computing MINIMUM MULTICUT is NP-hard even for unweighted binary trees [CFR03], but has a polynomial-time 2-approximation for trees [GVY97].

Consider a plane straight-line drawing of a tree $T = (V, E)$. We construct a tree T_M by subdividing every edge of T once as follows. Tree T_M has a vertex n_v for each vertex $v \in V$ and a vertex n_e for each edge $e \in E$. For each $e = uv \in E$, edges $n_u n_e$ and $n_e n_v$ are in T_M . The set X of terminal pairs contains a pair (n_e, n_f) for each pair of conflicting edges e, f of T . Let all edges of T_M have weight 1.

Lemma 6.5. *Let E' be a MINIMUM MULTICUT of T_M with respect to the terminal pairs X and let C_1^M, \dots, C_k^M denote the connected components of $T_M - E'$. Then, components $C_i = \{e \in E \mid n_e \in C_i^M\}$ form a minimum GRR decomposition of T with proper contacts.*

Proof. Consider a multicut E' of T_M , $|E'| = k - 1$. Consider a connected component C_i^M of $T_M - E'$. Then, the edges in $C_i = \{e \in E \mid n_e \in C_i^M\}$ are conflict-free and form a connected subtree T_i of T . Thus, T_i is a GRR by Lemma 6.2. It is easy to see that such GRRs have proper contacts. Consider a vertex $v \in V$ and edges $e, f, g, h \in E$, each incident to v , such that e and f are in one GRR and g and h in another. But then, n_e and n_f are in one connected component of $T_M - E'$ and n_g and n_h in another, a contradiction.

Next, consider a GRR decomposition of T into k subtrees $T_i = (V_i, E_i)$ with proper contacts. We create an edge set S iteratively as follows. Assume T_i, T_j touch at vertex $v \in V$. Let edge $e = uv$ be in T_i , and let v be a leaf in T_i . We then add edge $n_e n_v$ of T_M to set S , as long as the number of connected components in $T_M - S$ increases by one; see Figure 66a and 66b. In the end, we have $|S| = k - 1$. After removing S from T_M , no connected component contains vertices n_{e_1}, n_{e_2} for a pair of conflicting edges e_1, e_2 . Thus, S is a multicut.

We have shown that GRR decompositions of T with proper contacts and size k are in one-to-one correspondence with the multicuts of T_M of size $k - 1$. Therefore, minimum multicuts correspond to minimum GRR decompositions, and it follows that C_i form a minimum GRR decomposition of T with proper contacts. \square

Note that MINIMUM MULTICUT can be solved in polynomial time in directed trees [CLR03], i.e., trees whose edges can be directed such that for each terminal pair (s_i, t_i) , the s_i - t_i -path is directed. We note that this result cannot be applied in

our context, since we can get MINIMUM MULTICUT instances for which no such orientation is possible, see Figure 66b. However, using the approximation algorithm from [GVY97], we obtain the following result.

Corollary 6.1. *Given a plane straight-line drawing of a tree $T = (V, E)$, a partition of E into $2 \cdot \text{OPT} - 1$ increasing-chord subtrees of T having only proper contacts can be computed in time polynomial in n , where OPT is the minimum size of such a partition.*

6.4.2.2 Optimal solution

In the following we show how to find a minimum GRR partition with only non-crossing contacts in polynomial time. As is the case with minimum partitions of simple hole-free polygons into convex [CD85] or star-shaped [Kei85] components, our algorithm is based on dynamic programming. We describe the dynamic program in detail and use it to find minimum GTDs for the setting as in Section 6.4.2.1, as well as for the setting in which non-proper, but non-crossing contacts of GRRs are allowed. First, we shall prove the following theorem.

Theorem 6.3. *Given a plane straight-line drawing of a tree $T = (V, E)$, a partition of E into a minimum number of increasing-chord subtrees of T (minimum GTD) having only non-crossing contacts can be computed in time $O(n^6)$.*

At the end of Section 6.4.2.2, we modify our dynamic program slightly to prove Theorem 6.4, which shows the same result for the setting in which only partitions with proper contacts are considered.

Theorem 6.4. *Given a plane straight-line drawing of a tree $T = (V, E)$, a partition of E into a minimum number of increasing-chord subtrees of T (minimum GTD) having only proper contacts can be computed in time $O(n^6)$.*

subtree T_u

Let T be rooted. For each vertex u with parent π_u , let T_u be the subtree of u together with edge $\pi_u u$. We shall use the following definition.

root component

Definition 6.10 (root component). *Given a GRR partition of the edges of a rooted tree T' , we call all GRRs containing the root of T' the root components. If the root of T' has degree 1, every GRR partition of T' has one unique root component.*

A minimum partition is constructed from the solutions of subinstances as follows. Let u_1, \dots, u_d be the children of u . For subtrees T_{u_1}, \dots, T_{u_d} whose only common vertex is u , a minimum partition P' of $T' = \bigcup_i T_{u_i}$ induces partitions P_i of T_{u_i} . Furthermore, P' is created by choosing P_i as partitions of T_{u_i} and possibly merging some of the root components of T_{u_i} , $i = 1, \dots, d$. Note that P_i is not necessarily a minimum partition of T_{u_i} , if P_i allows us to merge more root components than a minimum partition of T_{u_i} would allow. Therefore, for every u we shall store minimum partitions of T_u for various possibilities of the root component of T_u . For the sake of uniformity, we choose a vertex with degree 1 as the root of T .

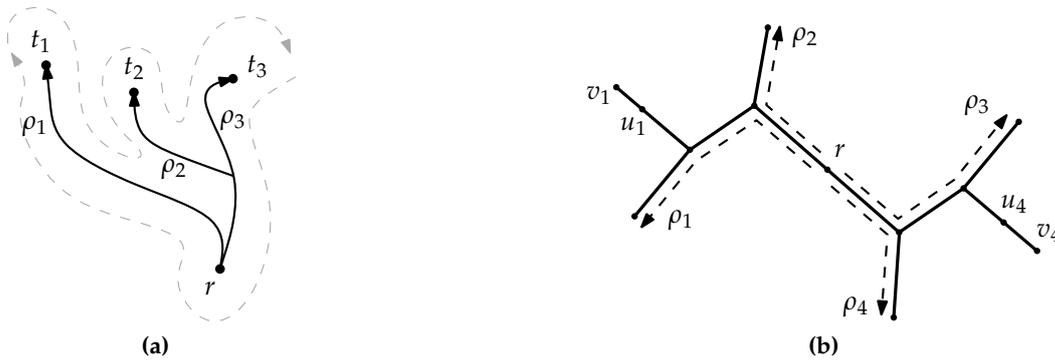


Figure 67: (a) Path ρ_2 is clockwise between paths ρ_1 and ρ_3 . (b) Statement of Lemma 6.6.

Given a tree root, the number of different subtrees it could be contained in may be exponential, e.g., it is $\Theta(2^n)$ in a star. The key observation for our algorithm is that we do not need to store a partition for each possible root component. We require the following notation.

Definition 6.11 (Path clockwise between). *Consider directed non-crossing paths ρ_1, ρ_2, ρ_3 with common origin r , endpoints t_1, t_2, t_3 and, possibly, common prefixes. Let V_i be vertices of ρ_i , $i = 1, 2, 3$, and let T be the tree formed by the union of ρ_1, ρ_2 and ρ_3 . We say that ρ_2 is clockwise between ρ_1 and ρ_3 , if the clockwise traversal of the outer face of T visits t_1, t_2, t_3 in this order; see Figure 67a.*

Note that in Definition 6.11 the three paths may (partially) coincide. Lemma 6.6 shows that to decide whether a union of two subtrees is increasing-chord, it is sufficient to consider only the two pairs of “outermost” root-leaf paths of each subtree. This result is crucial for limiting the number of representative decompositions that need to be considered during our dynamic programming approach. The statement of the lemma is illustrated in Figure 67b.

Lemma 6.6. *Let T_1, T_2 be increasing-chord trees sharing a single vertex r . Let all tree edges be directed away from r . Let paths ρ_1, ρ_2 in T_1 and ρ_3, ρ_4 in T_2 be paths from r to a leaf, such that:*

- every directed path from r in T_1 is clockwise between ρ_1 and ρ_2 ;
- every directed path from r in T_2 is clockwise between ρ_3 and ρ_4 ;
- for $i = 1, \dots, 4$, path ρ_i is clockwise between ρ_{i-1} and ρ_{i+1} (indices modulo 4).

Then, $\rho_1 \cup \rho_2 \cup \rho_3 \cup \rho_4$ is increasing-chord if and only if $T_1 \cup T_2$ is increasing-chord.

Proof. Consider trees T_1, T_2 and paths ρ_1, \dots, ρ_4 satisfying the condition of the lemma; see Figure 67b for a sketch. Note that ρ_1 and ρ_2 may have common prefixes, and so may ρ_3 and ρ_4 . Assume the four paths ρ_1, \dots, ρ_4 are drawn with increasing chords, but the union T' of the trees T_1 and T_2 is not. Then, without loss of generality, there exist edges u_1v_1 in T_1 and u_4v_4 in T_2 , such that the normal ℓ to u_1v_1 at u_1 crosses edge u_4v_4 .

As defined in Chapter 3, we say that an edge uv points upwards, downwards etc.



Figure 68: Constructions in the proof of Lemma 6.6.

Claim 2. *Without loss of generality, we may assume the following; see Figure 68.*

- (i) *Edge u_1v_1 points vertically upwards,*
- (ii) *edge u_4v_4 is the first edge on the $r-v_4$ path ρ'' crossed by ℓ and points upwards,*
- (iii) *vertex u_4 is on ℓ and to the right of u_1 .*

We ensure (i) by rotation. Then, point r is below ℓ (or on it), since the $r-v_1$ -path ρ' is increasing-chord. For (ii), we choose u_4v_4 as the first edge with this property. If it points downward, there is an edge on the $r-u_4$ -path crossed by ℓ . For (iii), if ℓ crosses u_4v_4 in an interior point p , we subdivide the edge at p and replace u_4v_4 by pv_4 . If u_4 is left of u_1 , we mirror the drawing horizontally. This proves the claim.

First, assume that v_1, v_4 are not on paths ρ_1, \dots, ρ_4 . Recall that two of the paths ρ_1, \dots, ρ_4 (without loss of generality, ρ_2 and ρ_3) are between ρ' and ρ'' . Let u_2v_2 and u_3v_3 be the last two edges on ρ_2 and ρ_3 , respectively. Note that $\text{ray}(u_1, v_1)$ and $\text{ray}(u_2, v_2)$ must diverge, and so must $\text{ray}(u_2, v_2)$ and $\text{ray}(u_3, v_3)$. If u_4v_4 points upwards and to the left as in Figure 68a, then $\text{ray}(u_3, v_3)$ and $\text{ray}(u_4, v_4)$ must converge; a contradiction. Thus, u_2v_2, u_3v_3 and u_4v_4 point upwards and to the right; see Figure 68b. Since T_1 as well as the union of ρ_1 and ρ_2 are increasing-chord, the angles $\angle v_1u_1u_2, \angle u_1u_2v_2, \angle v_2u_2u_3$ and $\angle u_2u_3v_3$ are between 90° and 180° . Therefore, vertices u_2 and u_3 must lie below ℓ . Let ℓ_3 be the normal to u_3v_3 at u_3 . Since T_2 is drawn with increasing chords, u_4v_4 must lie below ℓ_3 , a contradiction.

The proof works similarly if u_1v_1 is on ρ_2 (by identifying u_1v_1 and u_2v_2), and the remaining cases are symmetric. □

*leftmost and
rightmost paths*

We now describe our dynamic programs for proper and non-crossing contacts in detail. We first give an overview of the general approach, then describe the non-crossing case and afterwards modify it for proper contacts. For a root component R of T_u , let the *leftmost path* (or, respectively, the *rightmost path*) be the simple path in R starting at π_u which always chooses the next counterclockwise (clockwise) edge.

The basic idea of the dynamic program is as follows. For a given subtree T_u , we store the sizes of the minimum GTDs of T_u for different possibilities of the root component. We combine these solutions to compute minimum GTDs of bigger subtrees. For this step, we must be able to test which root components can be merged into one GRR. Instead of storing the partition sizes for *all* possible root components, we only store the minimum partition size for each combination of the leftmost and rightmost path of the root component. Thus, for each T_u , we only store $O(n^2)$ partition sizes. Note that this is sufficient, since by Lemma 6.6 the question whether two root components can be merged depends only on their leftmost and rightmost paths.

If u is the root of a subtree T' and has degree 2 or greater in T' , there might be several root components in a partition of T' , i.e., GRRs containing u . Let R be some fixed root component of the considered GTD. If u has degree 2 or greater in R , then we need a reference direction to define the leftmost and rightmost paths of R . Let ρ_l be the leftmost path of the rooted tree $R + \pi_u u$. Note that ρ_l contains the edge $\pi_u u$. Then, the leftmost path of R is $\rho_l - \pi_u u$. The rightmost path of R is defined analogously.

Recall that T_u is the subtree of u together with edge $\pi_u u$. For each pair of vertices t_i, t_j in T_u , cell $\tau[u, t_i, t_j]$ of a table τ stores the size of a minimum GRR decomposition of T_u , in which the root component has the $\pi_u t_i$ -path and the $\pi_u t_j$ -path as its leftmost and rightmost path, respectively. Cell $\tau[u]$ stores the size of a minimum GRR decomposition of T_u . We have $\tau[u] = \min_{t_i, t_j} \tau[u, t_i, t_j]$. For simplicity, we set $\min \emptyset = \infty$.

table τ
values $\tau[u, t_i, t_j]$

Clearly, for each leaf u , $\tau[u, u, u] = 1$, and $\tau[u, t_i, t_j] = \infty$ for all other values of t_i, t_j . Let v be the only neighbor of the root r of the tree T . Then, $\tau[v]$ is the size of a minimum GRR decomposition of T . We show how to compute τ bottom-up.

For ease of presentation, we use the following notation. Vertex u is not a leaf and has children u_1, \dots, u_d . Let π_u, u_1, \dots, u_d have this clockwise order around u . Let $t_i \neq u$ be a vertex in T_{u_i} . We define t_j, t_k, t_ℓ analogously for $1 \leq i \leq j \leq k \leq \ell \leq d$. Let ρ_i be the u - t_i -path.

We consider two settings: allowing arbitrary non-crossing contacts and allowing only proper contacts. The dynamic programs for the two cases are very similar, and the program for arbitrary non-crossing contacts is slightly more complex. To reduce duplication, we first present the program for arbitrary non-crossing contacts, and later show how to modify it for the case when only proper contacts are allowed.

6.4.2.3 Non-crossing contacts

Recall that vertex u can live in a root component R together with non-consecutive children $u_i, u_\ell, i < \ell$. If arbitrary non-crossing contacts are allowed, some nodes from $u_{i+1}, \dots, u_{\ell-1}$ that are not in R can also be in one GRR. Therefore, after choosing the root component R of T_u , we must be able to recursively compute the

minimum size of a partition of the union of T_{u_j} , $u_j \notin R$. We introduce additional tables for this purpose.

In addition to the table τ storing the values $\tau[u, t_i, t_j]$, we use tables σ_Δ for $\Delta = 1, \dots, 4$, as well as tables σ and σ_M . These additional tables will be used to formulate the recurrences for τ . For fixed u, i, j , the corresponding values of σ_Δ , σ and σ_M denote the sizes of minimum GTDs of $T_{u_i} \cup T_{u_{i+1}} \cup \dots \cup T_{u_j}$ with certain properties. Table σ_Δ considers different possibilities of the leftmost and rightmost paths of the root components as well as the degree Δ of u in the root component. Recall that in an increasing-chord tree drawing, every vertex has degree at most 4. Formally, the value $\sigma_\Delta[u, t_i, t_j]$ denotes the minimum number of GRRs in a GTD of the tree $T_{u_i} \cup T_{u_{i+1}} \cup \dots \cup T_{u_j}$, in which there exists a GRR R with the rightmost path $u-t_i$ and leftmost path $u-t_j$ and in which u has degree Δ in R .

$\sigma_\Delta[u, t_i, t_j]$

$\sigma[u, t_i, t_j]$

For some recurrences, we need to aggregate the various possibilities stored in σ_Δ . For this purpose, we use tables σ and σ_M as follows. The value σ is the minimum of σ_Δ over all values of Δ . We define $\sigma[u, t_i, t_j]$ as $\sigma[u, t_i, t_j] = \min_{\Delta=1, \dots, 4} \sigma_\Delta[u, t_i, t_j]$.

$\sigma_M[u, i, j]$

The value σ_M stores the minimum over all combinations of the leftmost and rightmost paths. Thus, it stores the size of the minimum partition of $T_{u_i} \cup \dots \cup T_{u_j}$, regardless of the root component. Formally, $\sigma_M[u, i, j]$ denotes the minimum number of GRRs in a GTD of $T_{u_i} \cup \dots \cup T_{u_j}$. Note that the arguments of $\sigma_M[u, \cdot, \cdot]$ are indices i, j of a pair of children of u , and the arguments of $\sigma_\Delta[u, \cdot, \cdot]$ and $\sigma[u, \cdot, \cdot]$ are a pair of vertices in $T_{u_i} \cup \dots \cup T_{u_j}$.

In the following recurrences, for a fixed pair of vertices t_i and t_ℓ , all possibilities for t_j and t_k are considered, such that both paths ρ_j and ρ_k are clockwise between ρ_i and ρ_ℓ . We test whether root components R_1 with the leftmost and rightmost paths ρ_i and ρ_j and R_2 with the leftmost and rightmost paths ρ_k and ρ_ℓ can be merged to a single GRR. We show that this covers all representative possibilities for a root component of a GTD of $T_{u_i} \cup \dots \cup T_{u_\ell}$ to have the leftmost and rightmost paths ρ_i and ρ_ℓ , respectively.

Lemma 6.7. *We have the recurrences*

- (1) $\sigma_1[u, t_i, t_j] = \sigma[u, t_i, t_j] = \tau[u_i, t_i, t_j]$ for all $t_i, t_j \neq u$ in T_{u_i} , $i = 1, \dots, d$;
- (2) $\sigma_M[u, i, i] = \tau[u_i]$ for all $i = 1, \dots, d$;
- (3) $\sigma_2[u, t_i, t_\ell] = \min_{t_j, t_k} \{ \sigma_1[u, t_i, t_j] + \sigma_M[u, j+1, k-1] + \sigma_1[u, t_k, t_\ell] - 1 \}$;
- (4) $\sigma_3[u, t_i, t_\ell] = \min_{t_j, t_k} \{ \min_{t_j, t_k} \{ \sigma_2[u, t_i, t_j] + \sigma_M[u, j+1, k-1] + \sigma_1[u, t_k, t_\ell] - 1 \}, \min_{t_j, t_k} \{ \sigma_1[u, t_i, t_j] + \sigma_M[u, j+1, k-1] + \sigma_2[u, t_k, t_\ell] - 1 \} \}$;
- (5) $\sigma_4[u, t_i, t_\ell] = \min_{t_j, t_k} \{ \sigma_1[u, t_i, t_i] + \sigma_M[u, i+1, j-1] + \sigma_1[u, t_j, t_j] + \sigma_M[u, j+1, k-1] + \sigma_1[u, t_k, t_k] + \sigma_M[u, k+1, \ell-1] + \sigma_1[u, t_\ell, t_\ell] \} - 3$.

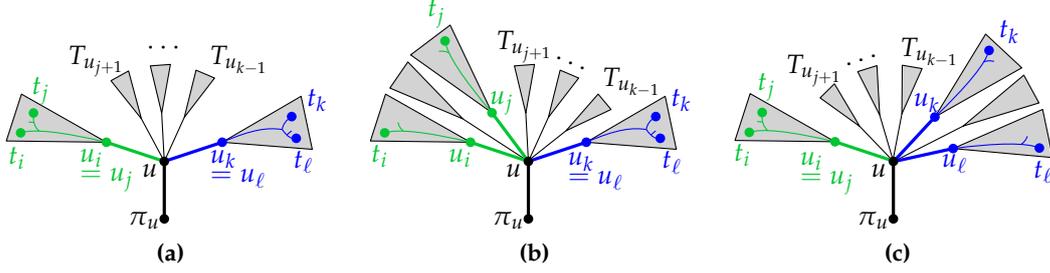


Figure 69: Recurrences in Lemma 6.7: (a) recurrence (3); (b) recurrence (4) for the case $m = j$; (c) recurrence (4) for the case $m = k$.

The minimizations in recurrences (3), (4) and (5) only consider vertices t_j, t_k , such that the subtree $\rho_i \cup \rho_j \cup \rho_k \cup \rho_\ell$ is increasing-chord.

Proof. Consider recurrence (1). First, we prove $\sigma_1[u, t_i, t_j] = \tau[u_i, t_i, t_j]$ as follows. For a GTD from the definition of $\sigma_1[u, t_i, t_j]$ we show that its size is an upper bound for $\tau[u_i, t_i, t_j]$. Then, for a GTD from the definition of $\tau[u_i, t_i, t_j]$, we show that its size is an upper bound for $\sigma_1[u, t_i, t_j]$.

Consider a GTD of $T_{u_i} \cup \dots \cup T_{u_j}$ of size x with root component R , such that R has $u-t_i$ and $u-t_j$ as its leftmost and rightmost paths, respectively. Since u has degree 1 in R , we have $i = j$. Thus, this partition is a GTD of T_{u_i} with R as the root component, so by definition of τ we have $\tau[u_i, t_i, t_j] \leq x$. Thus, we have $\sigma_1[u, t_i, t_j] \geq \tau[u_i, t_i, t_j]$. Conversely, consider a GTD of T_{u_i} , such that its root component R has $u-t_i$ and $u-t_j$ as its leftmost and rightmost paths. Thus, t_i and t_j are both in T_{u_i} , and vertex u has degree 1 in R . By the definition of σ_1 , this partition has size at least $\sigma_1[u, t_i, t_j]$. Thus, we have $\sigma_1[u, t_i, t_j] \leq \tau[u_i, t_i, t_j]$. Finally, since for $i = j$ we have $T_{u_i} \cup \dots \cup T_{u_j} = T_{u_i}$, vertex u can only have degree 1 in the root component of a GTD, so we have $\sigma_1[u, t_i, t_j] = \sigma[u, t_i, t_j]$. Thus, recurrence (1) holds.

Recurrence (2) holds trivially, since by the definitions of σ_M and $\tau[\cdot]$, both $\sigma_M[u, i, i]$ and $\tau[u_i]$ denote the size of the minimum GRR partition of T_{u_i} .

Consider recurrence (3) and a GTD P of $T_{u_i} \cup \dots \cup T_{u_\ell}$ of size x with root component R . Again, let R have $u-t_i$ and $u-t_\ell$ as its leftmost and rightmost paths, respectively. Let u have degree 2 in R . Therefore, $i \neq \ell$, and R only consists of two parts R_1, R_2 (green and blue in Figure 69a, respectively), such that R_1 is contained in T_{u_i} and R_2 is contained in T_{u_ℓ} . Partition P induces a GTD P_1 of T_{u_i} of size x_1 , a GTD P_2 of T_{u_ℓ} of size x_2 and a GTD P_3 of $T_{u_{j+1}} \cup \dots \cup T_{u_{\ell-1}}$ of size x_3 . Since $R_1 \cup R_2 = R$, we have $x = x_1 + x_2 + x_3 - 1$. Let u_j be a vertex in R_1 , such that $u-u_j$ is the rightmost path of R_1 . Let u_k be the vertex in R_2 , such that $u-u_k$ is the leftmost path of R_2 . The subtree $\rho_i \cup \rho_j \cup \rho_k \cup \rho_\ell$ is contained in R and, therefore, is increasing-chord. By the definition of σ_1 and σ_M , we have $\sigma_1[u, t_i, t_j] \leq x_1$, $\sigma_1[u, t_k, t_\ell] \leq x_2$ and $\sigma_M[u, j+1, k-1] \leq x_3$. Thus, the right part of recurrence (3) is at most x , so the right side is upper bounded by the left side.

Conversely, let the right side of recurrence (3) be less than ∞ . Let j, k, t_j, t_k be chosen such that the minimum on the right side is realized. Then, $\rho_i \cup \rho_j \cup \rho_k \cup \rho_\ell$ is increasing-chord. Let $\sigma_1[u, t_i, t_j] = x_1$, and let P_1 be a GTD of size x_1 realizing the minimum in the definition of $\sigma_1[u, t_i, t_j]$. Let R_1 be the root component of P_1 . Then, R_1 has leftmost and rightmost paths $u-t_i$ and $u-t_j$ respectively. Analogously, let $\sigma_1[u, t_k, t_\ell] = x_2$, and let P_2 be a GTD of size x_2 realizing the minimum in the definition of $\sigma_1[u, t_k, t_\ell]$. Let R_2 be the root component of P_2 . Then, R_2 has leftmost and rightmost paths $u-t_k$ and $u-t_\ell$ respectively. Finally, let P_3 be a GTD of size x_3 realizing the minimum in the definition of $\sigma_M[u, j+1, k-1]$. By Lemma 6.6, $R_1 \cup R_2$ is increasing-chord. Consider the GTD P formed by taking the union of P_1, P_2 and P_3 and merging R_1 and R_2 . Partition P has size $x_1 + x_2 + x_3 - 1$. Its root component R has leftmost and rightmost paths $u-t_i$ and $u-t_\ell$ respectively, and u has degree 2 in R . Thus, by the definition of $\sigma_2[u, t_i, t_\ell]$, we have $\sigma_2[u, t_i, t_\ell] \leq x_1 + x_2 + x_3 - 1$. Thus, the left side of recurrence (3) is upper bounded by its right side. Therefore, recurrence (3) holds.

Next, consider recurrence (4) and a GRR partition P of $T_{u_i} \cup \dots \cup T_{u_\ell}$ of size x with root component R . Once again, let R have $u-t_i$ and $u-t_\ell$ as its leftmost and rightmost paths, respectively. Let u have degree 3 in R . Therefore, $i \neq \ell$. In addition to u_i and u_ℓ , the GRR R must contain another child u_m of u , such that $i < m < \ell$. We can partition R into two GRRs R_1 and R_2 , such that u_i is in R_1 , u_ℓ in R_2 and u_m is either in R_1 or in R_2 . First, assume u_m is in R_1 ; see Figure 69b. The other case is symmetric; see Figure 69c. We choose $j = m$. Let t_j be a vertex in T_{u_j} , such that $u-t_j$ is the rightmost path of R_1 . Let t_k be a vertex in T_{u_ℓ} , such that $u-t_k$ is the leftmost path in R_2 . Note that in this case, t_k and t_ℓ are in the same subtree $T_{u_k} = T_{u_\ell}$. We can split the partition P into GRR partitions P_1 of $T_{u_i} \cup \dots \cup T_{u_j}$ of size x_1 , P_2 of T_{u_ℓ} of size x_2 and P_3 of $T_{u_{j+1}} \cup \dots \cup T_{u_{k-1}}$ of size x_3 . It holds: $R = R_1 \cup R_2$, and apart from R , no other GRR in P is split, since the contacts are non-crossing. Thus, $x = x_1 + x_2 + x_3 - 1$. By definition, $\sigma_2[u, t_i, t_j] \leq x_1$, $\sigma_1[u, t_k, t_\ell] \leq x_2$ and $\sigma_M[u, j+1, k-1] \leq x_3$. Therefore, the right side of recurrence (4) is at most x . The same holds for the symmetric case in which u_m is in R_2 by analogous arguments. Thus, the right side of recurrence (4) is upper bounded by its left side.

Conversely, let the right side of recurrence (4) be less than ∞ . Let j, k, t_j, t_k be chosen such that the minimum on the right side is realized. First, assume it is realized by $\sigma_2[u, t_i, t_j] + \sigma_M[u, j+1, k-1] + \sigma_1[u, t_k, t_\ell] - 1$. Then, $\rho_i \cup \rho_j \cup \rho_k \cup \rho_\ell$ is increasing-chord. Let $\sigma_2[u, t_i, t_j] = x_1$, and let P_1 be a GRR partition of size x_1 realizing the minimum in the definition of $\sigma_2[u, t_i, t_j]$. Let R_1 be the root component of P_1 . Then, R_1 has leftmost and rightmost paths $u-t_i$ and $u-t_j$ respectively. The degree of u in R_1 is 2, and the vertices t_i and t_j must lie in different subtrees T_{u_i} and T_{u_j} , respectively. Analogously, let $\sigma_1[u, t_k, t_\ell] = x_2$, and let P_2 be a GRR partition of size x_2 realizing the minimum in the definition of $\sigma_1[u, t_k, t_\ell]$. Let R_2 be the root component of P_2 . Then, R_2 has leftmost and rightmost paths $u-t_k$ and $u-t_\ell$ respectively. Finally, let P_3 be a GRR partition of size x_3 realizing the minimum in the definition of $\sigma_M[u, j+1, k-1]$. By Lemma 6.6, $R_1 \cup R_2$ is increasing-chord.

Consider the GRR partition P formed by taking the union of P_1 , P_2 and P_3 and merging R_1 and R_2 . Partition P has size $x_1 + x_2 + x_3 - 1$. Its root component R has leftmost and rightmost paths $u-t_i$ and $u-t_\ell$, respectively, and u has degree 3 in R . Therefore, by the definition of $\sigma_3[u, t_i, t_\ell]$, we have $\sigma_3[u, t_i, t_\ell] \leq x_1 + x_2 + x_3 - 1$. Thus, the left side of recurrence (4) is upper bounded by its right side. The same holds for the symmetric case in which the minimum on the right side is realized by $\sigma_1[u, t_i, t_j] + \sigma_M[u, j + 1, k - 1] + \sigma_2[u, t_k, t_\ell] - 1$. Therefore, recurrence (4) holds.

Finally, consider recurrence (5) and a GTD P of $T_{u_i} \cup \dots \cup T_{u_\ell}$ of size x with root component R . Once again, let R have $u-t_i$ and $u-t_\ell$ as its leftmost and rightmost paths, respectively. Let u have degree 4 in R . Then, R is a subdivision of $K_{1,4}$ [Ala+13]. Let t_j and t_k be the other two leaves of R lying in the subtrees T_{u_j} and T_{u_k} respectively, for $1 \leq i < j < k < \ell \leq d$. Then, we can split P into 7 GTDs P_1, \dots, P_7 as follows. Partitions P_1, P_2, P_3, P_4 are GTDs of subtrees $T_{u_i}, T_{u_j}, T_{u_k}$ and T_{u_ℓ} , respectively, with the respective sizes x_1, x_2, x_3, x_4 and paths $u-u_i, u-u_j, u-u_k$ and $u-u_\ell$ as the respective root components. Partitions P_5, P_6, P_7 are GTDs of $T_{u_{i+1}} \cup \dots \cup T_{u_{j-1}}, T_{u_{j+1}} \cup \dots \cup T_{u_{k-1}}$ and $T_{u_{k+1}} \cup \dots \cup T_{u_{\ell-1}}$, respectively, with respective sizes x_5, x_6 and x_7 . The root component R is split into the four paths $u-u_i, u-u_j, u-u_k$ and $u-u_\ell$, and no other GRR is split, since the contacts in P are non-crossing. Therefore, $x = x_1 + \dots + x_7 - 3$. By the definition of σ_1 , we have $\sigma_1[u, t_i, t_i] \leq x_1$, $\sigma_1[u, t_j, t_j] \leq x_2$, $\sigma_1[u, t_k, t_k] \leq x_3$ and $\sigma_1[u, t_\ell, t_\ell] \leq x_4$. By the definition of σ_M , $\sigma_M[u, i + 1, j - 1] \leq x_5$, $\sigma_M[u, j + 1, k - 1] \leq x_6$ and $\sigma_M[u, k + 1, \ell - 1] \leq x_7$. Thus, the right side of recurrence (5) is at most x , so the right side is upper bounded by the left side.

Conversely, let the right side of recurrence (5) be less than ∞ . Let j, k, t_j, t_k be chosen such that the minimum on the right side is realized. Then, $\rho_i \cup \rho_j \cup \rho_k \cup \rho_\ell$ is increasing-chord. Let $\sigma_1[u, t_i, t_i] = x_1$, $\sigma_1[u, t_j, t_j] = x_2$, $\sigma_1[u, t_k, t_k] = x_3$ and $\sigma_1[u, t_\ell, t_\ell] = x_4$. Let P_1, P_2, P_3 and P_4 be GTDs realizing the minimum in the definitions of $\sigma_1[u, t_i, t_i]$, $\sigma_1[u, t_j, t_j]$, $\sigma_1[u, t_k, t_k]$ and $\sigma_1[u, t_\ell, t_\ell]$, respectively. Next, let $\sigma_M[u, i + 1, j - 1] = x_5$, $\sigma_M[u, j + 1, k - 1] = x_6$ and $\sigma_M[u, k + 1, \ell - 1] = x_7$. Let P_5, P_6 and P_7 be GTDs realizing the minima in the definitions of $\sigma_M[u, i + 1, j - 1]$, $\sigma_M[u, j + 1, k - 1]$ and $\sigma_M[u, k + 1, \ell - 1]$, respectively. The four paths $\rho_i, \rho_j, \rho_k, \rho_\ell$ can be merged into a single GRR R with leftmost path ρ_i and rightmost path ρ_ℓ . Consider partition P with root component R formed by taking the union of P_1, \dots, P_7 and merging the four paths $\rho_i, \rho_j, \rho_k, \rho_\ell$. No more GRRs can be merged, since the contacts must be non-crossing. The GRR R is the root component of P . It has leftmost and rightmost paths $u-t_i$ and $u-t_\ell$ respectively, and u has degree 4 in R . Thus, by the definition of $\sigma_4[u, t_i, t_\ell]$, we have $\sigma_4[u, t_i, t_\ell] \leq x_1 + \dots + x_7 - 3$. Thus, the left side of recurrence (5) is upper bounded by its right side. Therefore, recurrence (5) holds. \square

Lemma 6.8. *We have the following recurrence.*

$$(6) \quad \sigma_M[u, i, \ell] = \min_{t_j, t_k} \{ \sigma_M[u, i, j - 1] + \sigma[u, t_j, t_k] + \sigma_M[u, k + 1, \ell] \}.$$

The minimization only considers j, k for $i \leq j \leq k \leq \ell$ and vertices t_j, t_k , such that t_j is in T_{u_j} and t_k is in T_{u_k} .

Proof. First, consider a GTD P of $T_{u_i} \cup \dots \cup T_{u_\ell}$. Consider a GRR R in P containing u with leftmost and rightmost paths $u-t_j$ and $u-t_k$, respectively, for some vertices t_j in T_{u_j} and t_k in T_{u_k} . Additionally, let R be chosen such that $k - j$ is maximized. Then, by the choice of R , no GRR in P has vertices both in $T_{u_i} \cup \dots \cup T_{u_{j-1}}$ and in $T_{u_{k+1}} \dots T_{u_\ell}$. Therefore, we can split partition P into GTDs P_1 of $T_{u_i} \cup \dots \cup T_{u_{j-1}}$ of size x_1 , P_2 of $T_{u_j} \cup \dots \cup T_{u_k}$ of size x_2 and P_3 of $T_{u_{k+1}} \cup \dots \cup T_{u_\ell}$ size x_3 , such that no GRR of P is split. Thus, $x = x_1 + x_2 + x_3$. By the definition of σ and σ_M , we have $\sigma_M[u, i, j-1] \leq x_1$, $\sigma[u, t_j, t_k] \leq x_2$ and $\sigma_M[u, k+1, \ell] \leq x_3$. Therefore, the right side of recurrence (6) is at most x , so the right side is upper bounded by the left side.

Conversely, let the right side of recurrence (6) be less than ∞ . Let j, k, t_j, t_k be chosen such that the minimum on the right side is realized. Let P_1, P_2, P_3 be GTDs of size x_1, x_2, x_3 , respectively, realizing the minima in the definitions of $\sigma_M[u, i, j-1]$, $\sigma[u, t_j, t_k]$ and $\sigma_M[u, k+1, \ell]$, respectively. The union of the three partitions is a GTD of $T_{u_i} \cup \dots \cup T_{u_\ell}$. Thus, by the definition of $\sigma_M[u, i, \ell]$, we have $\sigma_M[u, i, \ell] \leq x_1 + x_2 + x_3$, so the left side of recurrence (6) is upper bounded by its right side. Therefore, recurrence (6) holds. \square

Lemma 6.9. *We have the following recurrences regarding τ .*

$$(7) \quad \tau[u, u, u] = 1 + \sigma_M[1, d];$$

$$(8) \quad \tau[u, t_i, t_j] = \sigma_M[u, 1, i-1] + \sigma[u, t_i, t_j] + \sigma_M[u, j+1, d], \text{ if the subtree } \pi_u u + \rho_i \cup \rho_j \text{ is increasing-chord, and } \infty \text{ otherwise.}$$

In recurrence (8), vertex $t_i \neq u$ is in T_{u_i} and vertex $t_j \neq u$ is in T_{u_j} .

Proof. First, we prove recurrence (7). Let P be a GTD of $T_u = \pi_u u + T_{u_1} \cup \dots \cup T_{u_d}$, such that the edge $\pi_u u$ is the root component of P . Then, the other GRRs of P induce a partition P_1 of $T_{u_1} \cup \dots \cup T_{u_d}$. Let x_1 be the size of P_1 . Then, P has size $x_1 + 1$. Furthermore, by the definition of σ_M , $\sigma_M[u, 1, d] \leq x_1$. Thus, the right side of recurrence (7) is at most $x_1 + 1$, so the right side is upper bounded by the left side.

Conversely, let the right side of recurrence (7) be less than ∞ . Let P_1 be a GTD of $T_{u_1} \cup \dots \cup T_{u_d}$ size x_1 . We add edge $\pi_u u$ as a new GRR to P_1 and get a partition P of T_u of size $x_1 + 1$ having $\pi_u u$ as its root component. Thus, the left side of recurrence (7) is at most $x_1 + 1$, so the left side is upper bounded by the right side. Therefore, recurrence (7) holds.

We now prove recurrence (8). Let P be a GTD of T_u of size x with root component R , such that R has $\pi_u t_i$ and $\pi_u t_j$ as its leftmost and rightmost paths, respectively. Then, no GRR of P has edges both in $T_{u_1} \cup \dots \cup T_{u_{i-1}}$ and in $T_{u_{j+1}} \cup \dots \cup T_{u_d}$, since otherwise such a GRR would cross R . Thus, P can be split into GTDs P_1 of $T_{u_1} \cup \dots \cup T_{u_{i-1}}$ of size x_1 , P_2 of $\pi_u u + T_{u_i} \cup \dots \cup T_{u_j}$ of size x_2 and P_3 of $T_{u_{j+1}} \cup \dots \cup T_{u_d}$ of size x_3 , such that R is the root component of P_2 and such that we have $x = x_1 + x_2 + x_3$. By the definition of σ and σ_M , we have $\sigma_M[u, 1, i-1] \leq x_1$, $\sigma[u, t_i, t_j] \leq x_2$ and $\sigma_M[u, j+1, \ell] \leq x_3$. Thus, the right

side of recurrence (8) is at most x , so the right side is upper bounded by the left side.

Finally, let the right side of recurrence (8) be less than ∞ . Let P_1 be a GTD of $T_{u_1} \cup \dots \cup T_{u_{i-1}}$ of size x_1 , let P_2 be a GTD of $T_{u_i} \cup \dots \cup T_{u_j}$ of size x_2 and P_3 a GTD of $T_{u_{j+1}} \cup \dots \cup T_{u_d}$ of size x_3 , such that R is the root component of P_2 having leftmost and rightmost paths $u-t_i$ and $u-t_j$, respectively. If $\pi_u u + \rho_i \cup \rho_j$ is increasing-chord, by Lemma 6.6, the subtree $R_2 := \pi_u u + R$ is also a GRR. By taking the union of P_1, P_2 and P_3 and merging R and $\pi_u u$ into R_2 , we get a GTD P of T_u of size $x := x_1 + x_2 + x_3$ with the root component R_2 , such that R_2 has the leftmost and rightmost paths $\pi_u t_i$ and $\pi_u t_j$, respectively. By the definition of τ , we have $\tau[u, t_i, t_j] \leq x$, so the left side of recurrence (8) is upper bounded by the right side. Therefore, recurrence (8) holds. \square

We can now use the above recurrences to fill the tables $\tau, \sigma, \sigma_\Delta$ and σ_M in polynomial time. This proves Theorem 6.3.

Theorem 6.3. Given a plane straight-line drawing of a tree $T = (V, E)$, a partition of E into a minimum number of increasing-chord subtrees of T (minimum GTD) having only *non-crossing* contacts can be computed in time $O(n^6)$.

Proof. For each pair $s, t \in V$, it can be tested in time $O(n)$ whether the path $s-t$ is increasing-chord [Ala+13]. We store the result for each pair $s, t \in V$, which allows us to query in time $O(1)$ whether any $s-t$ -path is increasing-chord. This precomputation takes $O(n^3)$ time.

We process the vertices $u \in V$ bottom-up and fill the tables $\tau[u, \cdot, \cdot], \sigma[u, \cdot, \cdot], \sigma_\Delta[u, \cdot, \cdot]$ and $\sigma_M[u, \cdot, \cdot]$. Consider a vertex $u \in V$ and assume all these values have been computed for all successors of u .

Using recurrences (1) and (2), we can compute all values of $\sigma_1[u, t_i, t_j]$ and $\sigma_M[u, i, i]$ in $O(n^2)$ time. We compute the remaining values $\sigma_\Delta[u, t_i, t_\ell], \sigma[u, t_i, t_\ell]$ and $\sigma_M[u, i, \ell]$ by an induction over $\ell - i$. For a fixed $m \geq 0$, assume all these values have been computed for $\ell - i \leq m$. We show how to compute them for $\ell - i = m + 1$.

First, we compute the new values $\sigma_\Delta[u, t_i, t_\ell]$ from the already computed ones using recurrences (3), ..., (6). This can be done in $O(n^4)$ time by testing all combinations of t_i, t_j, t_k, t_ℓ . Next, we compute $\sigma[u, t_i, t_\ell] = \min_{\Delta=1, \dots, 4} \sigma_\Delta[u, t_i, t_\ell]$ in $O(n^2)$ time. After that, the new values $\sigma_M[u, i, \ell]$ can be computed using recurrence (6). This can be done in $O(n^4)$ time by testing all combinations of i, ℓ, t_j, t_k .

In this way, we compute all values $\sigma_\Delta[u, t_i, t_\ell], \sigma[u, t_i, t_\ell]$ and $\sigma_M[u, i, \ell]$, for all $\ell - i \leq d$, in $O(n^5)$ time. Then, we compute $\tau[u, t_i, t_j]$ using recurrences (7) and (8). This can be done in $O(n^2)$ time by testing all combinations of t_i and t_j . After that, we compute $\tau[u]$. It took us $O(n^5)$ time to compute all the values for the vertex u .

Let r be the root of T , and let v be the only child of r . By the above procedure, we can compute $\tau[v]$ in $O(n^6)$ time. Since $T = T_v$, $\tau[v]$ is the minimum size of a GTD of T . \square

For partitions allowing edge splits, we use the results from Section 6.2.2 to reduce the problem to the scenario without edge splits.

Corollary 6.2. *An optimal partition of a plane straight-line tree drawing into GRRs with non-crossing contacts can be computed in $O(n^6)$ time, if no edge splits are allowed, and in $O(n^{12})$ time, if edge splits are allowed.*

6.4.2.4 Proper contacts

For GTDs allowing only proper contacts of GRRs, we can modify the above dynamic program. We redefine $\sigma_M[u, i, j]$ to be the size of a minimum GTD of $T_{u_i} \cup \dots \cup T_{u_j}$, in which no two edges uu_i, \dots, uu_j are in the same GRR. Furthermore, we replace two recurrences as follows.

Lemma 6.10. *For GTDs with proper contacts, the following recurrences replace recurrences (6) and (7).*

$$(6') \sigma_M[u, i, j] = \sum_{m=i}^j \sigma_1[u, m, m];$$

$$(7') \tau[u, u, u] = 1 + \min_{t_i, t_j} \{ \sigma_M[u, 1, i-1] + \sigma[u, t_i, t_j] + \sigma_M[u, j+1, d] \}.$$

The minimization in recurrence (7') only considers i, j for $1 \leq i \leq j \leq d$ and vertices t_i, t_j , such that t_i is in T_{u_i} and t_j is in T_{u_j} .

Recurrence (6') follows trivially from the new definition of σ_M . The proof of recurrence (7') is very similar to the proof of Lemma 6.8. Recurrences (1), ..., (5) and (8) still hold and can be proved by reusing the proofs of Lemma 6.7 and 6.9. The runtime of the modified dynamic program remains the same. This proves Theorem 6.4.

Theorem 6.4. Given a plane straight-line drawing of a tree $T = (V, E)$, a partition of E into a minimum number of increasing-chord subtrees of T (minimum GTD) having only proper contacts can be computed in time $O(n^6)$.

Analogously as for non-crossing contacts, we use the results from Section 6.2.2 to extend the result to GTDs allowing edge splits.

Corollary 6.3. *An optimal partition of a plane straight-line tree drawing into GRRs with proper contacts can be computed in $O(n^6)$ time, if no edge splits are allowed, and in $O(n^{12})$ time, if edge splits are allowed.*

Note that Corollary 6.3 provides a better runtime than the dynamic program in the conference version of this work [NPR15].

6.5 TRIANGULATIONS

In this section, we consider GRR partitions of polygonal regions. Recall that a polygonal region is a GRR if and only if it contains no pairs of conflicting edges. Further, recall that GRRs that are polygonal regions need not be convex

redefining $\sigma_M[u, i, j]$

and that they do not have holes [TK12]. Since partitioning polygonal regions into a minimum number of GRRs is NP-hard [TK12], we study special cases of this problem.

We consider partitioning a hole-free polygon \mathcal{P} with a fixed triangulation into a minimum number of GRRs by cutting it along chords of \mathcal{P} contained in the triangulation. For such decompositions we restrict the GRRs to consist of a group of triangles of the triangulation whose union forms a simple polygon without articulation points. Note that allowing articulation points makes the problem NP-hard. To prove this, we can easily turn the plane straight-line tree drawing Γ from Section 6.4.1, which is a subdivision of a star, into a hole-free triangulated polygon with a single articulation point corresponding to the star center.

We reduce the problem to MINIMUM MULTICUT on trees and use it to give a polynomial-time $(2 - 1/\text{OPT})$ -approximation, where OPT is the number of GRRs in an optimal partition. Let Δ_{uvw} be the triangle defined by three non-collinear points u, v, w .

 Δ_{uvw}

Lemma 6.11. *Let \mathcal{P} be a simple polygon, uv an edge on its boundary and $w \notin \mathcal{P}$ another point, such that $\mathcal{P} \cap \Delta_{uvw} = uv$. If \mathcal{P} is not a GRR, neither is $\mathcal{P} \cup \Delta_{uvw}$.*

Proof. Polygon $\mathcal{P}' = \mathcal{P} \cup \Delta_{uvw}$ can become a GRR only if uv is a conflict edge in \mathcal{P} . Then, either uv is crossed by a normal ray to another edge, or a normal ray to uv crosses another edge. In the former case, either uw or wv is crossed by a normal ray to another edge, a contradiction to the greediness of $\mathcal{P} \cup \Delta_{uvw}$.

In the latter case, there exists a point p in the interior of uv , such that $\text{ray}_{uv}(p)$ crosses the boundary $\partial\mathcal{P}$ of \mathcal{P} . Let y be the first intersection point; see Figure 70a. Then, either $\text{ray}_{uv}(u)$ or $\text{ray}_{uv}(v)$ must also cross $\partial\mathcal{P}$. Without loss of generality, there exists a point x on $\partial\mathcal{P}$, such that: vx and uv are orthogonal, $vx \cap \mathcal{P} = \{v, x\}$, and adding edge vx to \mathcal{P} would create an inner face f , such that u is not on the boundary of f ; see Figure 70a.

Let ρ be the v - x -path on the boundaries of both \mathcal{P} and f . Without loss of generality, let uv point upwards, and let x lie to the right of uv . Then, w must lie to the right of the line through uv , and there must exist a point q on vw , such that $\text{ray}_{vw}(q)$ intersects ρ . \square

From now on, let triangles τ_1, \dots, τ_n form a triangulation of a simple hole-free polygon \mathcal{P} , and let T be its corresponding dual binary tree. For simplicity we use τ_i to refer both to a triangle in \mathcal{P} and its dual node in T .

Definition 6.12 (Projection of an edge). *For three non-collinear points u_1, u_2, u_3 , let $\text{proj}_{u_1}(u_2u_3)$ denote the set of points covered by shifting u_2u_3 orthogonally to itself and away from u_1 (blue in Figure 70b).*

Definition 6.13 (Conflicting triangles). *Let $\tau_i = \Delta_{u_1u_2u_3}$ and $\tau_j = \Delta_{v_1v_2v_3}$ be two triangles such that the two edges dual to u_1u_2 and v_1v_2 are on the τ_i - τ_j -path in T . We call τ_i, τ_j conflicting, if $\text{proj}_{u_1}(u_2u_3) \cup \text{proj}_{u_2}(u_1u_3)$ contains an interior point of τ_j or, symmetrically, $\text{proj}_{v_1}(v_2v_3) \cup \text{proj}_{v_2}(v_1v_3)$ contains an interior point of τ_i .*

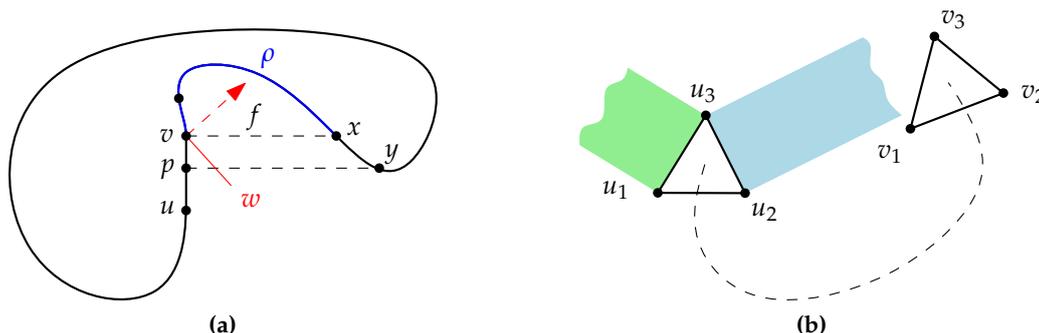


Figure 70: (a) Construction for the proof of Lemma 6.11. When adding triangles as in the lemma, \mathcal{P} remains non-greedy. (b) Conflicting triangles.

Lemma 6.12. *Let $T' \subset T$ be a subtree of T and let \mathcal{P}' be the corresponding simple polygon dual to T' . Then \mathcal{P}' is a GRR if and only if no two triangles τ, τ' in \mathcal{P}' are conflicting.*

Proof. Assume there are two conflicting triangles $\tau_i = \triangle_{u_1u_2u_3}, \tau_j = \triangle_{v_1v_2v_3}$ in T' . Let \mathcal{P}'' denote the polygon defined by the τ_i - τ_j -path in T' and assume that the two edges dual to u_1u_2 and v_1v_2 are on the τ_i - τ_j -path. Since τ_i and τ_j are conflicting, there is, without loss of generality, a point p on u_2u_3 such that $\text{ray}_{u_2u_3}(p)$ intersects an edge of τ_j . Hence, \mathcal{P}'' is not a GRR. Moreover, \mathcal{P}' is obtained from \mathcal{P}'' by adding triangles. Thus Lemma 6.11 implies that \mathcal{P}' cannot be a GRR.

Conversely, assume \mathcal{P}' is not a GRR. There exists an outer edge uv of \mathcal{P}' and a point x in the interior of uv such that $\text{ray}_{uv}(x)$ crosses another boundary edge of \mathcal{P}' in a point y . Let τ_x, τ_y be the triangles with $x \in \tau_x$ and $y \in \tau_y$. Then τ_x and τ_y are conflicting. \square

By Lemma 6.12, the decompositions of \mathcal{P} in k GRRs correspond bijectively to the multicuts E' of T with $|E'| = k - 1$ where the terminal pairs are the pairs of conflicting triangles.

We now use the 2-approximation for MINIMUM MULTICUT on trees [GVY97] to give a $(2 - 1/\text{OPT})$ -approximation for the minimum GRR decomposition of \mathcal{P} . Let E' be a 2-approximation of MINIMUM MULTICUT in T with respect to the pairs of conflicting triangles. By the above observation the minimum multicut for T has size $\text{OPT} - 1$, hence $|E'| \leq 2\text{OPT} - 2$, which in turn yields a decomposition into $2\text{OPT} - 1$ regions. Thus, the approximation guarantee is $2 - 1/\text{OPT}$. We summarize this in Theorem 6.5.

Theorem 6.5. *There is a polynomial-time $(2 - 1/\text{OPT})$ -approximation for minimum GRR decomposition of triangulated simple polygons.*

6.6 HEURISTICS FOR SIMPLE POLYGONS

We implemented three heuristics for partitioning simple polygons into a minimum number of GRRs.

1. Heuristics **B** corresponds to the algorithm by Tan and Kermarrec [TK12]. In the first phase, we start with a single component. We then iteratively cut every component that is not a GRR by a bisector ray from a maximum concave angle. We repeat until every component is a GRR. In the second phase, we iteratively merge pairs of adjacent components, if the resulting component is a GRR. The pairs are chosen arbitrarily. We stop when no such pair exists.
2. Heuristics **T**. First, we triangulate the input polygon using a constrained Delaunay triangulation (CDT). We then merge pairs of adjacent regions analogously to the second phase of heuristics **B**.
3. Heuristics **M**. Again, we first triangulate the input polygon using a CDT. For this triangulation, we formulate an instance of MINIMUM MULTICUT on a tree using Lemma 6.12, which we then solve exactly using an integer linear program (ILP).

We used the CDT implementation from the CGAL library [Yvi17] and the Gurobi solver [GUR7.5] for the ILP.

In terms of partition size, heuristics **M** is at least as good as **T**, since the result of **T** corresponds to an inclusion-minimal (but not necessarily minimum) multicut in the setting of Lemma 6.12. Note that for some polygons, heuristics **B** and **T** can perform arbitrarily worse than heuristics **M**; see Figure 51. In our experiments, both **B** and **T** were outperformed by **M**, whereas heuristics **B** and **T** provided partitions of similar size; see examples in Figures 71, 72 and 73.

6.7 CONCLUSION

Motivated by a geographic routing algorithm for dense wireless sensor networks proposed by Tan and Kermarrec [TK12], we further studied the problem of finding minimum GRR decompositions of polygons. We considered the special case of decomposing plane straight-line drawings of graphs, which correspond to infinitely thin polygons. For this case, we could apply insights gained from the study of self-approaching and increasing-chord drawings by the graph drawing community.

We extended the result of Tan and Kermarrec [TK12] for polygonal regions with holes by showing that partitioning a plane graph drawing into a minimum number of increasing-chord components is NP-hard. We then considered plane drawings of trees and showed how to model the decomposition problem using MINIMUM MULTICUT, which provided a polynomial-time 2-approximation. We solved the partitioning problem for trees optimally in polynomial time using dynamic programming. Finally, using insights gained from the decomposition of graph drawings, we gave a polynomial-time 2-approximation for decomposing triangulated polygons along their chords.

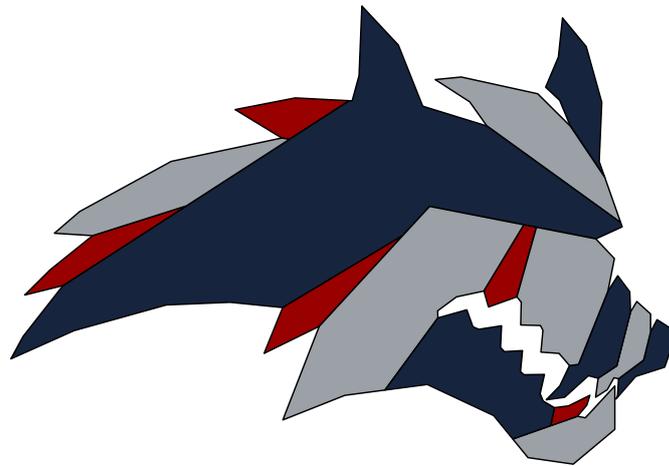
Open questions

For the NP-hard problem of decomposing plane drawings of graphs into the minimum number of GRRs, it is interesting to find approximation algorithms.

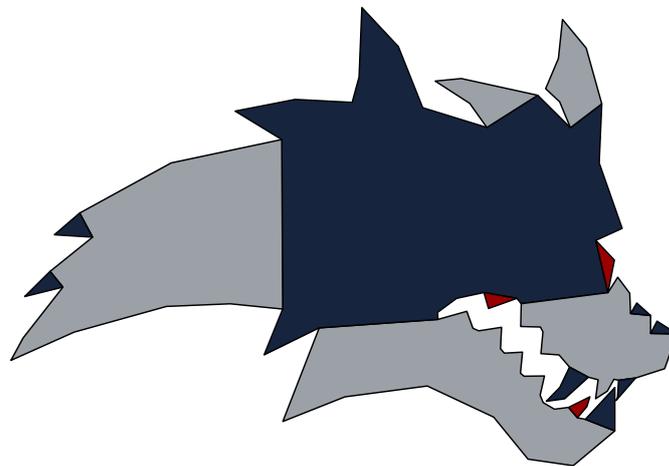
For decomposing polygons, many problems remain open. For example, one could investigate whether minimum decomposition is NP-hard for simple polygons for different types of allowed partitions. Is finding the optimum solution hard for partitioning triangulations as in Section 6.5? Is the minimum GRR decomposition problem hard if we allow cutting the polygon at any diagonal? Is it hard if arbitrary polygonal cuts are allowed, i.e., the partition can use Steiner points? Finally, are there approximations for partitioning polygons with and without holes into GRRs?

Acknowledgements

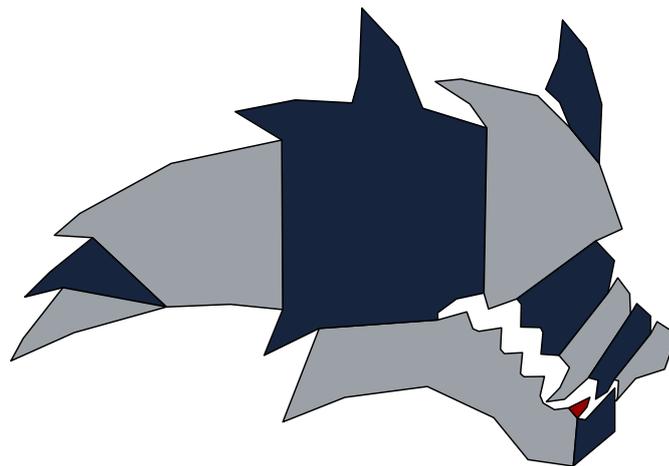
The author thanks Jie Gao for pointing him to the topic of GRR decompositions.



(a) heuristics B: 16 components

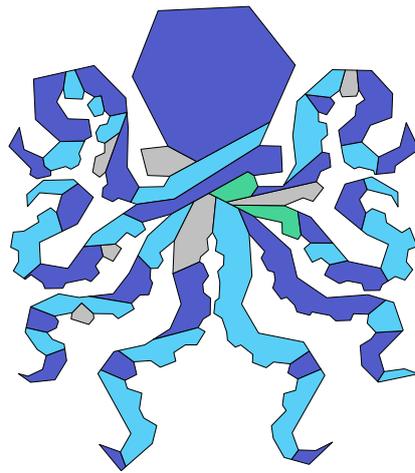


(b) heuristics T: 16 components

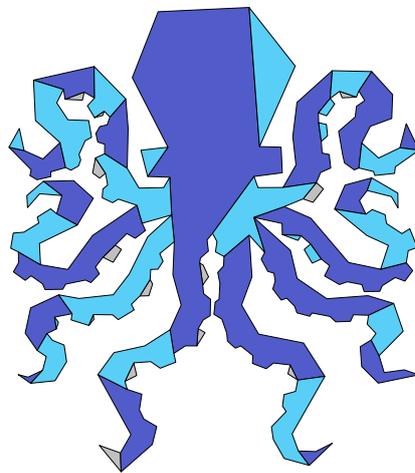


(c) heuristics M: 13 components

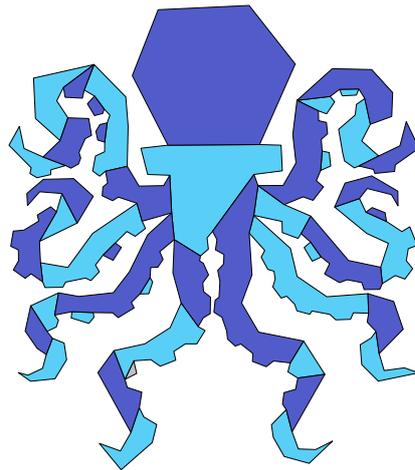
Figure 71: Comparing heuristics for GRR partition. The input polygon has 100 vertices and is inspired by the Stony Brook Seawolves logo (<http://www.stonybrookathletics.com>).



(a) heuristics B: 57 components



(b) heuristics T: 54 components



(c) heuristics M: 46 components

Figure 72: Comparing heuristics for GRR partition. The input polygon has 347 vertices and is an adaptation of an image downloaded from http://www.vikinganswerlady.com/Stars/Heraldry_SVG_Images/Fish.htm.

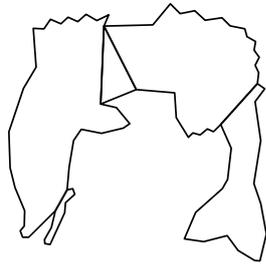
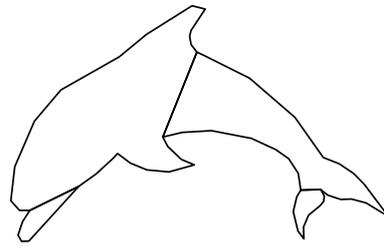
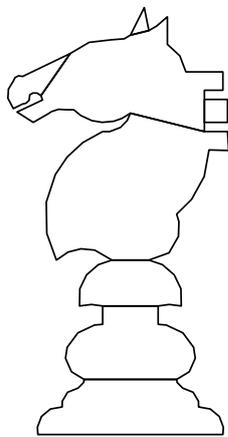
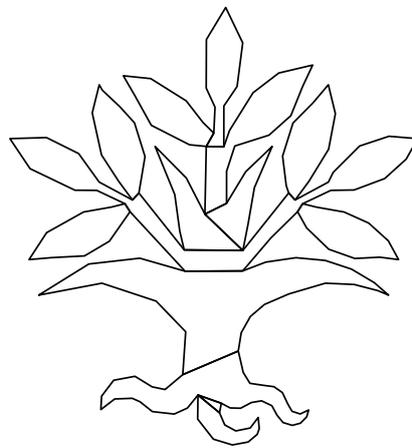
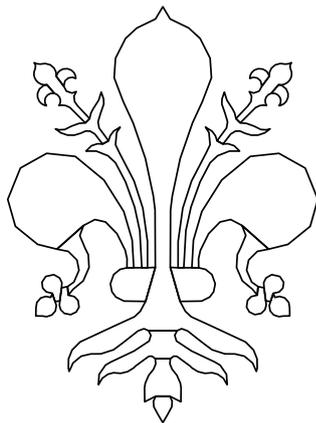
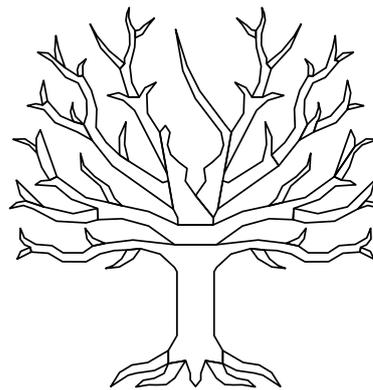
(a) **B**: 7, **T**: 7, **M**: 5(b) **B**: 5, **T**: 4, **M**: 4(c) **B**: 10, **T**: 13, **M**: 8(d) **B**: 17, **T**: 18, **M**: 14(e) **B**: 37, **T**: 33, **M**: 26(f) **B**: 50, **T**: 58, **M**: 46

Figure 73: Comparing GRR partition sizes for heuristics **B**, **T** and **M** on six additional instances. The figures show the results of heuristics **M**. The input polygons are adaptations of images downloaded from http://www.vikinganswerlady.com/Stars/Heraldry_SVG_Images/index.htm.

CONCLUSION

In this thesis, I have studied the following questions related to greedy routing on geometrically embedded graphs. 1) Which graphs admit a greedy, self-approaching or increasing-chord embedding? 2) What is the complexity of partitioning plane graph drawings and polygons into a small number of components that support greedy routing?

Results from Chapters 4 and 5 advance the state of knowledge regarding the first question. In Chapter 4, I have given the first characterization of trees that admit a greedy embedding. Previous research on the existence of greedy embeddings mainly focused on 3-connected planar graphs and binary cactus graphs. Characterizing greedy-drawable trees is an important next step for tackling the remaining graph classes.

For self-approaching and increasing-chord drawings, the characterization of trees admitting such drawings has been the only result so far. In Chapter 5, I have continued this line of research for other popular and important graph classes. For example, I have shown existence of increasing-chord drawings for triangulations as well as non-existence of self-approaching drawings for families of binary cactus graphs.

Self-approaching and increasing-chord graph drawings have become a popular research topic in the graph drawing and computational geometry community, which lets one hope that new existence results for the remaining graph classes will follow in the near future.

In Chapter 6, I have extensively studied the complexity of partitioning plane graph drawings into a minimum number of Greedily Routable Regions and have shown NP-hardness even for partitioning drawings of trees, whereas natural restrictions thereof turned out to be optimally solvable in polynomial time. For partitioning polygons, no such optimal solutions or even polynomial-time approximations have been known. The insights gained from studying the GRR partitioning problem for graph drawings have allowed me to formulate a polynomial-time 2-approximation for partitioning hole-free polygons with respect to a given triangulation.

CONSEQUENCES FOR PRACTICE

For greedy drawings, negative existence results [PR05; LM10] as well as worst-case exponential area bounds [ADF12] have been known for trees, which leads one to believe that the vision of using Euclidean greedy embeddings for point-to-point routing is unrealistic for the case when the communication graph is weakly connected. The negative existence results for trees as well as worst-case exponential area bounds for binary cactuses acquired in this thesis support this belief. Some possibilities to overcome these issues are 1) efficient distributed embedding techniques that do not provide a 100% delivery guarantee of the standard greedy routing [Rao+03; Sar+09], 2) network decomposition techniques similar to the one described in Chapter 6 or 3) non-Euclidean greedy embeddings [Kle07].

Similarly, for self-approaching and increasing-chord graph drawings, the negative existence result as well as the worst-case exponential area bound for binary cactuses shown in Chapter 5 of this thesis supports the belief that these drawing styles are not suited for creating practical drawings of big weakly connected graphs. Instead, one could consider using graph drawing styles with weaker notions of geodesic-path tendency that are less restrictive, such as monotone drawings [Ang+12], and study the possibilities to further improve the practicality and aesthetics of such drawings.

OUTLOOK

For each of the considered problems, several remaining open questions are listed at the end of the corresponding chapter. The strong Papadimitriou and Ratajczak conjecture that 3-connected planar graphs have planar *convex* graph drawings is particularly worth mentioning here in the light of the recent proof by Da Lozzo et al. [DDF17] that such graphs have *planar* greedy drawings.

In the context of greedy embeddings, an interesting research direction is to consider algorithmic problems that model aspects of wireless sensor networks that are important for the practical applicability for routing. Such aspects that have been considered are, for example, succinctness [EG11; GS09] and efficient distributed computation [Sar+09; Ben+11] of virtual coordinates, bounded hop stretch [FPW09] and load balancing [Sar+10]. However, combining these aspects while maintaining provable guarantees appears challenging. Another challenge in this context is node mobility and edge removal due to battery depletions. The following problem posed by Kleinberg [Kle07] remains open: design a distributed greedy embedding algorithm that allows reconstructing the greedy embedding upon node or edge deletion in $O(\text{polylog } n)$ amortized time or prove that $o(n)$ amortized time is unachievable.

To use greedy, strongly monotone, self-approaching and increasing-chord drawings as a tool to help users find paths in graphs, the drawings should satisfy certain aesthetic criteria such as a low number of crossings and a good resolution. It has been known that greedy drawings of trees require exponential resolution

in the worst case [ADF12], and I have shown this for strongly monotone tree drawings in Chapter 5. For higher connectivity, e.g., 3-connected graphs, it is still open whether there exist graphs that require exponential area for any greedy, self-approaching or increasing-chord drawing. For greedy drawings, this open problem has been stated by Angelini et al. [ADF12]. What about the area requirements for *planar* greedy, self-approaching or increasing-chord drawings of 3-connected planar graphs? Another possible research direction is considering combinations with well-established graph drawing conventions such as orthogonal graph drawings. In a joint work with Angelini et al. [Ang+18], we have recently initiated the study of greedy rectilinear orthogonal drawings.

We have seen that minimum GRR partition is computationally hard even for very restricted variants of the problem. For partitioning simple polygons and polygonal regions into a minimum number of GRRs, a possible direction of future work is designing and comparing new efficient heuristics. Of particular interest are algorithms that can be efficiently implemented in a distributed setting.

Part III

APPENDIX

A

APPENDIX

Appendix A contains the proof of a sufficient condition for $\{180^\circ, \varphi_1, \dots, \varphi_4\} \in \mathcal{P}^5$ deferred from Chapter 4.

Lemma 4.25. *Consider angles $0^\circ \leq \varphi_4 \leq 60^\circ$, $90^\circ < \varphi_3 \leq \varphi_2 \leq \varphi_1 \leq 120^\circ$, $\varphi_1 + \dots + \varphi_4 > 360^\circ$. Let the following two conditions hold:*

- (i) $14\varphi_1 + 12\varphi_2 + 8\varphi_3 + 15\varphi_4 > 4500^\circ$
- (ii) For $x := \min\{\frac{1}{7}(14\varphi_1 + 12\varphi_2 + 8\varphi_3 + 15\varphi_4 - 4500^\circ), \varphi_4\}$ and $p_1 \in [0^\circ, 90^\circ]^{10}$, $p_1 = (\beta_0, \dots, \beta_4, \gamma_0, \dots, \gamma_4)$ defined as:

$$\begin{aligned}\beta_0 &= \varphi_4 - x, \\ \beta_1 &= 90^\circ - \frac{x}{2}, \\ \beta_2 &= \varphi_3 + \frac{\varphi_2}{2} + \frac{\varphi_1}{4} + \frac{\varphi_4 - x}{8} - 157.5^\circ, \\ \beta_3 &= \varphi_2 + \frac{\varphi_1}{2} + \frac{\varphi_4 - x}{4} - 135^\circ, \\ \beta_4 &= \varphi_1 - 90^\circ + \frac{\varphi_4 - x}{2}, \\ \gamma_0 &= 90^\circ - \frac{\varphi_4 - x}{2}, \\ \gamma_1 &= x, \\ \gamma_2 &= 168.75^\circ - \frac{\varphi_3}{2} - \frac{\varphi_2}{4} - \frac{\varphi_1}{8} - \frac{\varphi_4 - x}{16}, \\ \gamma_3 &= 157.5^\circ - \frac{\varphi_2}{2} - \frac{\varphi_1}{4} - \frac{\varphi_4 - x}{8}, \\ \gamma_4 &= 135^\circ - \frac{\varphi_1}{2} - \frac{\varphi_4 - x}{4},\end{aligned}$$

it holds: $\omega(p_1) < 0$.

Then, $\{180^\circ, \varphi_1, \dots, \varphi_4\} \in \mathcal{P}^5$.

Proof. Assume both conditions hold. See the construction in Figure 74. The angles in p_1 are chosen such that all five triangles are equilateral: we have $\beta_1 = \alpha_1$

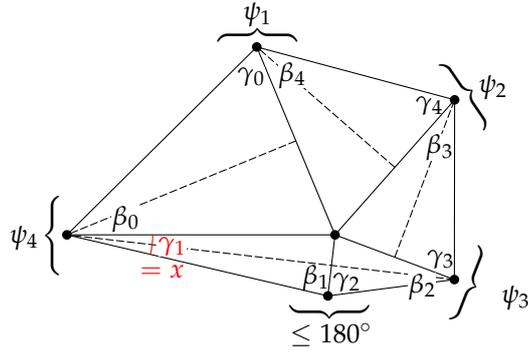


Figure 74: Proof of Lemma 4.25.

and $\gamma_i = \alpha_i$ for $i = 0, 2, 3, 4$. Furthermore, $\beta_0, \gamma_1, \beta_2, \beta_3, \beta_4 \leq 60^\circ$. Consider the permutation $\tau = (4, 0, 3, 2, 1)$. We verify the conditions in Observation 4.3:

$$\begin{aligned} \beta_4 + \gamma_0 &= \varphi_1 \\ \beta_3 + \gamma_4 &= \varphi_2 \\ \beta_2 + \gamma_3 &= \varphi_3 \\ \beta_1 + \gamma_2 &= 258.75^\circ - \frac{1}{8}\varphi_1 - \frac{1}{4}\varphi_2 - \frac{1}{2}\varphi_3 - \frac{1}{16}\varphi_4 - \frac{7}{16}x \leq 180^\circ \\ \beta_0 + \gamma_1 &= \varphi_4 \\ 2\beta_0 + \gamma_0 &= \frac{3}{2}\varphi_4 - \frac{3}{2}x + 90^\circ \leq 180^\circ \\ 2\beta_1 + \gamma_1 &= 180^\circ \\ 2\beta_2 + \gamma_2 &= \frac{3}{8}\varphi_1 + \frac{3}{4}\varphi_2 + \frac{3}{2}\varphi_3 + \frac{3}{16}\varphi_4 - \frac{3}{16}x - 146.25^\circ \leq 180^\circ \\ 2\beta_3 + \gamma_3 &= \frac{3}{4}\varphi_1 + \frac{3}{2}\varphi_2 + \frac{3}{8}\varphi_4 - \frac{3}{8}x - 112.5^\circ \leq 180^\circ \\ 2\beta_4 + \gamma_4 &= \frac{3}{2}\varphi_1 + \frac{3}{4}\varphi_4 - \frac{3}{4}x - 45^\circ \leq 180^\circ \\ \beta_0 + 2\gamma_0 &= 180^\circ \\ \beta_1 + 2\gamma_1 &= 90^\circ + \frac{3}{2}x \leq 180^\circ \\ \beta_2 + 2\gamma_2 &= 180^\circ \\ \beta_3 + 2\gamma_3 &= 180^\circ \\ \beta_4 + 2\gamma_4 &= 180^\circ \end{aligned}$$

We see that p_1 lies in the polytope $P_\tau =: P$, in particular, $p_1 \in \partial P$.

We now define another point $p_2 \in [0^\circ, 90^\circ]^{10}$. Due to condition (i), there must exist $90^\circ < \psi_i < \varphi_i$ for $i = 1, \dots, 3$, $0^\circ < \psi_4 < \varphi_4$, $0^\circ < \varepsilon < \psi_4$ (for proper ψ_i , ε can be chosen arbitrarily small), such that:

$$14\psi_1 + 12\psi_2 + 8\psi_3 + 15\psi_4 - 80\varepsilon = 4500^\circ.$$

Consider the point $p_2 = (\bar{\beta}_0, \dots, \bar{\beta}_4, \bar{\gamma}_0, \dots, \bar{\gamma}_4)$, such that:

$$\begin{aligned}\bar{\beta}_0 &= \psi_4, \\ \bar{\beta}_1 &= 90^\circ - \varepsilon, \\ \bar{\beta}_2 &= \psi_3 + \frac{\psi_2}{2} + \frac{\psi_1}{4} + \frac{\psi_4}{8} - 157.5^\circ, \\ \bar{\beta}_3 &= \psi_2 + \frac{\psi_1}{2} + \frac{\psi_4}{4} - 135^\circ, \\ \bar{\beta}_4 &= \psi_1 - 90^\circ + \frac{\psi_4}{2}, \\ \bar{\gamma}_0 &= 90^\circ - \frac{\psi_4}{2} - \varepsilon, \\ \bar{\gamma}_1 &= 0^\circ, \\ \bar{\gamma}_2 &= 168.75^\circ - \frac{\psi_3}{2} - \frac{\psi_2}{4} - \frac{\psi_1}{8} - \frac{\psi_4}{16} - \varepsilon, \\ \bar{\gamma}_3 &= 157.5^\circ - \frac{\psi_2}{2} - \frac{\psi_1}{4} - \frac{\psi_4}{8} - \varepsilon, \\ \bar{\gamma}_4 &= 135^\circ - \frac{\psi_1}{2} - \frac{\psi_4}{4} - \varepsilon.\end{aligned}$$

The condition $\sum_{i=0}^4 (\bar{\beta}_i + \bar{\gamma}_i) = 540^\circ$ holds, since

$$\begin{aligned}16 \sum_{i=0}^4 (\bar{\beta}_i + \bar{\gamma}_i) &= 16 \cdot (90^\circ + 168.75^\circ) + (14\psi_1 + 12\psi_2 + 8\psi_3 + 15\psi_4 - 16 \cdot 5\varepsilon) \\ &= 16 \cdot 540^\circ\end{aligned}$$

due to the choice of ψ_i and ε . The rest of the conditions for $p_2 \in P$ can be easily verified:

$$\begin{aligned}\bar{\beta}_4 + \bar{\gamma}_0 &= \psi_1 - \varepsilon < \varphi_1 \\ \bar{\beta}_3 + \bar{\gamma}_4 &= \psi_2 - \varepsilon < \varphi_2 \\ \bar{\beta}_2 + \bar{\gamma}_3 &= \psi_3 - \varepsilon < \varphi_3 \\ \bar{\beta}_1 + \bar{\gamma}_2 &= 258.75^\circ - \frac{1}{8}\psi_1 - \frac{1}{4}\psi_2 - \frac{1}{2}\psi_3 - \frac{1}{16}\psi_4 - 2\varepsilon < 180^\circ \\ \bar{\beta}_0 + \bar{\gamma}_1 &= \psi_4 < \varphi_4 \\ 2\bar{\beta}_0 + \bar{\gamma}_0 &= \frac{3}{2}\psi_4 + 90^\circ - \varepsilon < 180^\circ \\ 2\bar{\beta}_1 + \bar{\gamma}_1 &= 180^\circ - 2\varepsilon < 180^\circ \\ 2\bar{\beta}_2 + \bar{\gamma}_2 &= \frac{3}{8}\psi_1 + \frac{3}{4}\psi_2 + \frac{3}{2}\psi_3 + \frac{3}{16}\psi_4 - \varepsilon - 146.25^\circ < 180^\circ \\ 2\bar{\beta}_3 + \bar{\gamma}_3 &= \frac{3}{4}\psi_1 + \frac{3}{2}\psi_2 + \frac{3}{8}\psi_4 - \varepsilon - 112.5^\circ < 180^\circ \\ 2\bar{\beta}_4 + \bar{\gamma}_4 &= \frac{3}{2}\psi_1 + \frac{3}{4}\psi_4 - \varepsilon - 45^\circ < 180^\circ\end{aligned}$$

$$\overline{\beta}_0 + 2\overline{\gamma}_0 = 180^\circ - 2\varepsilon < 180^\circ$$

$$\overline{\beta}_1 + 2\overline{\gamma}_1 = 90^\circ - \varepsilon < 180^\circ$$

$$\overline{\beta}_2 + 2\overline{\gamma}_2 = 180^\circ - 2\varepsilon < 180^\circ$$

$$\overline{\beta}_3 + 2\overline{\gamma}_3 = 180^\circ - 2\varepsilon < 180^\circ$$

$$\overline{\beta}_4 + 2\overline{\gamma}_4 = 180^\circ - 2\varepsilon < 180^\circ$$

Apart from $\overline{\gamma}_1 \geq 0^\circ$, all inequalities are strict. Since $\gamma_1 > 0^\circ$, for each $\lambda \in (0, 1)$, the point $\lambda p_1 + (1 - \lambda)p_2$ lies in the interior of P . Since $\omega(p_1) < 0$ and $\omega(p_2) > 0$ (due to $\overline{\gamma}_1 = 0^\circ$), by the mean value theorem, $\omega(\lambda p_1 + (1 - \lambda)p_2) = 0$ for some $\lambda \in (0, 1)$. \square

BIBLIOGRAPHY

- [ACM89] Esther M. Arkin, Robert Connelly, and Joseph S. B. Mitchell. „On Monotone Paths Among Obstacles with Applications to Planning Assemblies.“ In: *Proc. 5th Ann. ACM Symp. Computational Geometry (SoCG'89)*. ACM, 1989, 334–343. DOI: [10.1145/73833.73870](https://doi.org/10.1145/73833.73870).
- [ADF12] Patrizio Angelini, Giuseppe Di Battista, and Fabrizio Frati. „Succinct Greedy Drawings Do Not Always Exist.“ In: *Networks* 59.3 (2012), 267–274. DOI: [10.1002/net.21449](https://doi.org/10.1002/net.21449).
- [AFG10] Patrizio Angelini, Fabrizio Frati, and Luca Grilli. „An Algorithm to Construct Greedy Drawings of Triangulations.“ In: *J. Graph Algorithms Appl.* 14.1 (2010), 19–51. DOI: [10.7155/jgaa.00197](https://doi.org/10.7155/jgaa.00197).
- [Aic+01] Oswin Aichholzer, Franz Aurenhammer, Christian Icking, Rolf Klein, Elmar Langetepe, and Günter Rote. „Generalized Self-Approaching Curves.“ In: *Discrete Appl. Math.* 109 (1–2 2001), 3–24. DOI: [10.1016/S0166-218X\(00\)00233-X](https://doi.org/10.1016/S0166-218X(00)00233-X).
- [AK04] Jamal N. Al-Karaki and Ahmed E. Kamal. „Routing Techniques in Wireless Sensor Networks: A Survey.“ In: *IEEE Wireless Communications* 11.6 (2004), 6–28. DOI: [10.1109/MWC.2004.1368893](https://doi.org/10.1109/MWC.2004.1368893).
- [Ala+13] Soroush Alamdari, Timothy M. Chan, Elyot Grant, Anna Lubiw, and Vinayak Pathak. „Self-Approaching Graphs.“ In: *Proc. 20th Internat. Symp. Graph Drawing (GD'12)*. Ed. by Walter Didimo and Maurizio Patrignani. Vol. 7704 of *Lecture Notes Comput. Sci.* Springer, 2013, 260–271. DOI: [10.1007/978-3-642-36763-2_23](https://doi.org/10.1007/978-3-642-36763-2_23).
- [AM12] Eiman Alotaibi and Biswanath Mukherjee. „A Survey on Routing Algorithms for Wireless Ad-Hoc and Mesh Networks.“ In: *Computer Networks* 56.2 (2012), 940–965. DOI: [10.1016/j.comnet.2011.10.011](https://doi.org/10.1016/j.comnet.2011.10.011).
- [Ang+12] Patrizio Angelini, Enrico Colasante, Giuseppe Di Battista, Fabrizio Frati, and Maurizio Patrignani. „Monotone Drawings of Graphs.“ In: *J. Graph Algorithms Appl.* 16.1 (2012), 5–35. DOI: [10.7155/jgaa.00249](https://doi.org/10.7155/jgaa.00249).
- [Ang+15] Patrizio Angelini, Walter Didimo, Stephen G. Kobourov, Tamara Mchedlidze, Vincenzo Roselli, Antonios Symvonis, and Stephen Wismath. „Monotone Drawings of Graphs with Fixed Embedding.“ In: *Algorithmica* 71.2 (2015), 233–257. DOI: [10.1007/s00453-013-9790-3](https://doi.org/10.1007/s00453-013-9790-3).
- [Ang+18] Patrizio Angelini, Michael A. Bekos, Walter Didimo, Luca Grilli, Philipp Kindermann, Tamara Mchedlidze, Roman Prutkin, Antonios Symvonis, and Alessandra Tappini. „Greedy Rectilinear Orthogonal Drawings.“ Under submission. 2018.

- [Ang17] Patrizio Angelini. „Monotone Drawings of Graphs with Few Directions.“ In: *Information Processing Letters* 120 (2017), 16–22. DOI: [10.1016/j.ipl.2016.12.004](https://doi.org/10.1016/j.ipl.2016.12.004).
- [AV10] Ian F. Akyildiz and Mehmet Can Vuran. *Wireless Sensor Networks*. John Wiley & Sons, 2010. DOI: [10.1002/9780470515181](https://doi.org/10.1002/9780470515181).
- [Bah+17] Yeganeh Bahoo, Stephane Durocher, Sahar Mehrpour, and Debajyoti Mondal. „Exploring Increasing-Chord Paths and Trees.“ In: *Proc. 29th Canadian Conf. Computational Geometry (CCCG'17)*. 2017, 19–24. arXiv: [1702.08380](https://arxiv.org/abs/1702.08380).
- [Bar66] David Barnette. „Trees in Polyhedral Graphs.“ In: *Canadian J. Math.* 18 (1966), 731–736. DOI: [10.4153/CJM-1966-073-4](https://doi.org/10.4153/CJM-1966-073-4).
- [Ben+11] Mirela Ben Chen, Steven J. Gortler, Craig Gotsman, and Camille Wormser. „Distributed Computation of Virtual Coordinates for Greedy Routing in Sensor Networks.“ In: *Discrete Appl. Math.* 159.7 (2011), 544–560. DOI: [10.1016/j.dam.2010.10.016](https://doi.org/10.1016/j.dam.2010.10.016).
- [BGJ07] Jehoshua Bruck, Jie Gao, and Anxiao (Andrew) Jiang. „MAP: Medial Axis Based Geometric Routing in Sensor Networks.“ In: *Wireless Networks* 13.6 (2007), 835–853. DOI: [10.1007/s11276-006-9857-z](https://doi.org/10.1007/s11276-006-9857-z).
- [Bir+11] Michael Biro, Jie Gao, Justin Iwerks, Irina Kostitsyna, and Joseph S. B. Mitchell. „Beacon-Based Routing and Coverage.“ In: *Proc. 21st Fall Workshop on Computational Geometry (FWCG'11)*. 2011.
- [Bir+13] Michael Biro, Justin Iwerks, Irina Kostitsyna, and Joseph S. B. Mitchell. „Beacon-Based Algorithms for Geometric Routing.“ In: *Proc. 13th Internat. Symp. Algorithms and Data Structures (WADS'13)*. Ed. by Frank Dehne, Roberto Solis-Oba, and Jörg-Rüdiger Sack. Vol. 8037 of *Lecture Notes Comput. Sci.* Springer, 2013, 158–169. DOI: [10.1007/978-3-642-40104-6_14](https://doi.org/10.1007/978-3-642-40104-6_14).
- [Bir13] Michael Biro. „Beacon-Based Routing and Guarding.“ PhD thesis. State University of New York at Stony Brook, Department of Applied Mathematics and Statistics, 2013.
- [BKL17] Prosenjit Bose, Irina Kostitsyna, and Stefan Langerman. „Self-approaching paths in simple polygons.“ In: *Proc. 33rd Internat. Symp. Computational Geometry (SoCG'17)*. Ed. by Boris Aronov and Matthew J. Katz. Vol. 77 of *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017, 21:1–21:15. DOI: [10.4230/LIPIcs.SocG.2017.21](https://doi.org/10.4230/LIPIcs.SocG.2017.21).
- [BMo4] Prosenjit Bose and Pat Morin. „Online Routing in Triangulations.“ In: *SIAM J. Comput.* 33.4 (2004), 937–951. DOI: [10.1137/S0097539700369387](https://doi.org/10.1137/S0097539700369387).

- [Bon+16] Nicolas Bonichon, Prosenjit Bose, Paz Carmi, Irina Kostitsyna, Anna Lubiw, and Sander Verdonschot. „Gabriel Triangulations and Angle-Monotone Graphs: Local Routing and Recognition.“ In: *Proc. 24th Internat. Symp. Graph Drawing and Network Visualization (GD'16)*. Ed. by Yifan Hu and Martin Nöllenburg. Vol. 9801 of *Lecture Notes Comput. Sci.* Springer, 2016, 519–531. DOI: [10.1007/978-3-319-50106-2_40](https://doi.org/10.1007/978-3-319-50106-2_40).
- [Bon+17] Nicolas Bonichon, Prosenjit Bose, Jean-Lou Carufel, Ljubomir Perković, and André van Renssen. „Upper and Lower Bounds for Online Routing on Delaunay Triangulations.“ In: *Discrete Comput. Geom.* 58.2 (2017), 482–504. DOI: [10.1007/s00454-016-9842-y](https://doi.org/10.1007/s00454-016-9842-y).
- [Bos+01] Prosenjit Bose, Pat Morin, Ivan Stojmenović, and Jorge Urrutia. „Routing with Guaranteed Delivery in Ad Hoc Wireless Networks.“ In: *Wireless Networks* 7.6 (2001), 609–616. DOI: [10.1023/A:1012319418150](https://doi.org/10.1023/A:1012319418150).
- [Bos+02] Prosenjit Bose, Andrej Brodnik, Svante Carlsson, Erik D. Demaine, Rudolf Fleischer, Alejandro López-Ortiz, Pat Morin, and J. Ian Munro. „Online Routing in Convex Subdivisions.“ In: *Internat. J. Comput. Geom. Appl.* 12.4 (2002), 283–295. DOI: [10.1142/S021819590200089X](https://doi.org/10.1142/S021819590200089X).
- [Bos+12] Prosenjit Bose, Rolf Fagerberg, André van Renssen, and Sander Verdonschot. „Competitive Routing in the Half- Θ_6 -Graph.“ In: *Proc. 23rd ACM-SIAM Symp. Discrete Algorithms (SODA'12)*. SIAM, 2012, 1319–1328. DOI: [10.1137/1.9781611973099.104](https://doi.org/10.1137/1.9781611973099.104).
- [Bos+15] Prosenjit Bose, Stephane Durocher, Debajyoti Mondal, Maxime Peabody, Matthew Skala, and Mohammad Abdul Wahid. „Local Routing in Convex Subdivisions.“ In: *Proc. 41st Internat. Conf. Current Trends in Theory and Practice of Computer Science (SOFSEM'15)*. Ed. by Giuseppe F. Italiano, Tiziana Margaria-Steffen, Jaroslav Pokorný, Jean-Jacques Quisquater, and Roger Wattenhofer. Vol. 8939 of *Lecture Notes Comput. Sci.* Springer, 2015, 140–151. DOI: [10.1007/978-3-662-46078-8_12](https://doi.org/10.1007/978-3-662-46078-8_12).
- [Bou+08] Azzedine Boukerche, Mohammad Z. Ahmad, Damla Turgut, and Begumhan Turgut. „A Taxonomy of Routing Protocols in Sensor Networks.“ In: *Algorithms and Protocols for Wireless Sensor Networks*. Ed. by Azzedine Boukerche. John Wiley & Sons, 2008, 129–160. DOI: [10.1002/9780470396360.ch6](https://doi.org/10.1002/9780470396360.ch6).
- [Bou08] Azzedine Boukerche, ed. *Algorithms and Protocols for Wireless Sensor Networks*. John Wiley & Sons, 2008. DOI: [10.1002/9780470396360](https://doi.org/10.1002/9780470396360).
- [BS13] Prosenjit Bose and Michiel Smid. „On Plane Geometric Spanners: A Survey and Open Problems.“ In: *Comput. Geom. Theory Appl.* 46.7 (2013), 818–830. DOI: [10.1016/j.comgeo.2013.04.002](https://doi.org/10.1016/j.comgeo.2013.04.002).
- [CA06] Qing Cao and Tarek Abdelzaher. „Scalable Logical Coordinates Framework for Routing in Wireless Sensor Networks.“ In: *ACM Trans. Sensor Networks* 2.4 (2006), 557–593. DOI: [10.1145/1218556.1218561](https://doi.org/10.1145/1218556.1218561).

- [Car+05] Antonio Caruso, Stefano Chessa, Swades De, and Alessandro Urpi. „GPS Free Coordinate Assignment and Routing in Wireless Sensor Networks.“ In: *Proc. 24th IEEE Internat. Conf. Computer Communications (INFOCOM'05)*. IEEE, 2005, 150–160. DOI: [10.1109/INFCOM.2005.1497887](https://doi.org/10.1109/INFCOM.2005.1497887).
- [CD85] Bernard Chazelle and David P. Dobkin. „Optimal Convex Decompositions.“ In: *Computational Geometry*. Ed. by Godfried T. Toussaint. Vol. 2 of *Machine Intelligence and Pattern Recognition*. North-Holland, 1985, 63–133. DOI: [10.1016/B978-0-444-87806-9.50009-8](https://doi.org/10.1016/B978-0-444-87806-9.50009-8).
- [CFR03] Gruia Călinescu, Cristina G. Fernandes, and Bruce Reed. „Multicuts in Unweighted Graphs and Digraphs with Bounded Degree and Bounded Tree-Width.“ In: *J. Algorithms* 48.2 (2003), 333–359. DOI: [10.1016/S0196-6774\(03\)00073-7](https://doi.org/10.1016/S0196-6774(03)00073-7).
- [Cha15] Nessrine Chakchouk. „A Survey on Opportunistic Routing in Wireless Communication Networks.“ In: *IEEE Communications Surveys and Tutorials* 17.4 (2015), 2214–2241. DOI: [10.1109/COMST.2015.2411335](https://doi.org/10.1109/COMST.2015.2411335).
- [Che14] Shiri Chechik. „Compact Routing Schemes.“ In: *Encyclopedia of Algorithms*. Ed. by Ming-Yang Kao. Springer, 2014, 1–3. DOI: [10.1007/978-3-642-27848-8_561-1](https://doi.org/10.1007/978-3-642-27848-8_561-1).
- [Che89] L. Paul Chew. „There Are Planar Graphs Almost as Good as the Complete Graph.“ In: *J. Computer and System Sciences* 39.2 (1989), 205–219. DOI: [10.1016/0022-0000\(89\)90044-5](https://doi.org/10.1016/0022-0000(89)90044-5).
- [ĆKW15] Ante Ćustić, Bettina Klinz, and Gerhard J. Woeginger. „Geometric Versions of the Three-Dimensional Assignment Problem under General Norms.“ In: *Discrete Optimization* 18 (2015), 38–55. DOI: [10.1016/j.disopt.2015.07.002](https://doi.org/10.1016/j.disopt.2015.07.002).
- [CL03] Bennett Chow and Feng Luo. „Combinatorial Ricci Flows on Surfaces.“ In: *J. Differential Geom.* 63.1 (2003), 97–129. DOI: [10.4310/jdg/1080835659](https://doi.org/10.4310/jdg/1080835659).
- [CLR03] Marie-Christine Costa, Lucas Létocart, and Frédéric Roupin. „A Greedy Algorithm for Multicut and Integral Multiflow in Rooted Trees.“ In: *Operations Research Letters* 31.1 (2003), 21–27. DOI: [10.1016/S0167-6377\(02\)00184-0](https://doi.org/10.1016/S0167-6377(02)00184-0).
- [CLR05] Marie-Christine Costa, Lucas Létocart, and Frédéric Roupin. „Minimal Multicut and Maximal Integer Multiflow: A Survey.“ In: *European J. Operational Research* 162.1 (2005), 55–69. DOI: [10.1016/j.ejor.2003.10.037](https://doi.org/10.1016/j.ejor.2003.10.037).
- [Com00] Douglas E. Comer. *Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architectures*. 4th edition. Prentice Hall, 2000.

- [DDF17] Giordano Da Lozzo, Anthony D'Angelo, and Fabrizio Frati. „On planar greedy drawings of 3-connected planar graphs.“ In: *Proc. 33rd Internat. Symp. Computational Geometry (SoCG'17)*. Ed. by Boris Aronov and Matthew J. Katz. Vol. 77 of *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017, 33:1–33:16. DOI: [10.4230/LIPIcs.SocG.2017.33](https://doi.org/10.4230/LIPIcs.SocG.2017.33).
- [DFG15] Hooman Reisi Dehkordi, Fabrizio Frati, and Joachim Gudmundsson. „Increasing-Chord Graphs on Point Sets.“ In: *J. Graph Algorithms Appl.* 19.2 (2015), 761–778. DOI: [10.7155/jgaa.00348](https://doi.org/10.7155/jgaa.00348).
- [DFS90] David P. Dobkin, Steven J. Friedman, and Kenneth J. Supowit. „Delaunay Graphs Are Almost as Good as Complete Graphs.“ In: *Discrete Comput. Geom.* 5.4 (1990), 399–407. DOI: [10.1007/BF02187801](https://doi.org/10.1007/BF02187801).
- [DGW15] Stephane Durocher, Leszek Gaşieniec, and Prudence W.H. Wong. „Routing in Geometric Networks.“ In: *Encyclopedia of Algorithms*. Ed. by Ming-Yang Kao. Springer, 2015, 1–5. DOI: [10.1007/978-3-642-27848-8_352-2](https://doi.org/10.1007/978-3-642-27848-8_352-2).
- [Dha10] Raghavan Dhandapani. „Greedy Drawings of Triangulations.“ In: *Discrete Comput. Geom.* 43.2 (2010), 375–392. DOI: [10.1007/s00454-009-9235-6](https://doi.org/10.1007/s00454-009-9235-6).
- [DKN10] Stephane Durocher, David Kirkpatrick, and Lata Narayanan. „On Routing with Guaranteed Delivery in Three-Dimensional Ad Hoc Wireless Networks.“ In: *Wireless Networks* 16.1 (2010), 227–235. DOI: [10.1007/s11276-008-0126-1](https://doi.org/10.1007/s11276-008-0126-1).
- [DKR15] Søren Dahlgaard, Mathias Bæk Tejs Knudsen, and Noy Rotbart. „A Simple and Optimal Ancestry Labeling Scheme for Trees.“ In: *Proc. 42nd Internat. Colloquium on Automata, Languages, and Programming (ICALP'15)*. Ed. by Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann. Vol. 9135 of *Lecture Notes Comput. Sci.* Springer, 2015, 564–574. DOI: [10.1007/978-3-662-47666-6_45](https://doi.org/10.1007/978-3-662-47666-6_45).
- [DS05] David M. Doolin and Nicholas Sitar. „Wireless Sensors for Wildfire Monitoring.“ In: *Proc. SPIE 12th Int. Symp. Smart Structures and Materials*. Vol. 5765 of *SPIE*. 2005, 477–484. DOI: [10.1117/12.605655](https://doi.org/10.1117/12.605655).
- [DV96] Giuseppe Di Battista and Luca Vismara. „Angles of Planar Triangular Graphs.“ In: *SIAM J. Discrete Math.* 9.3 (1996), 349–359. DOI: [10.1137/S0895480194264010](https://doi.org/10.1137/S0895480194264010).
- [EG11] David Eppstein and Michael T. Goodrich. „Succinct Greedy Geometric Routing Using Hyperbolic Geometry.“ In: *IEEE Trans. Computers* 60.11 (2011), 1571–1580. DOI: [10.1109/TC.2010.257](https://doi.org/10.1109/TC.2010.257).
- [Epp00] David Eppstein. „Spanning Trees and Spanners.“ In: *Handbook of Computational Geometry*. Ed. by Jörg-Rüdiger Sack and Jorge Urrutia. North-Holland, 2000, 425–461. DOI: [10.1016/B978-044482537-7/50010-3](https://doi.org/10.1016/B978-044482537-7/50010-3).

- [Fan+05] Qing Fang, Jie Gao, Leonidas J. Guibas, Vin de Silva, and Li Zhang. „GLIDER: Gradient Landmark-Based Distributed Routing for Sensor Networks.“ In: *Proc. 24th IEEE Internat. Conf. Computer Communications (INFOCOM'05)*. Vol. 1. IEEE, 2005, 339–350. DOI: [10.1109/INFCOM.2005.1497904](https://doi.org/10.1109/INFCOM.2005.1497904).
- [Fel+16] Stefan Felsner, Alexander Igamberdiev, Philipp Kindermann, Boris Klemz, Tamara Mchedlidze, and Manfred Scheucher. „Strongly Monotone Drawings of Planar Graphs.“ In: *Proc. 32nd Internat. Symp. Computational Geometry (SoCG'16)*. Ed. by Sándor Fekete and Anna Lubiw. Vol. 51 of *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2016, 37:1–37:15. DOI: [10.4230/LIPIcs.SoCG.2016.37](https://doi.org/10.4230/LIPIcs.SoCG.2016.37).
- [Felo4] Stefan Felsner. „Schnyder Woods or How to Draw a Planar Graph?“ In: *Geometric Graphs and Arrangements. Advanced Lectures in Mathematics*. Vieweg Verlag, 2004, 17–42. DOI: [10.1007/978-3-322-80303-0_2](https://doi.org/10.1007/978-3-322-80303-0_2).
- [Fin87] Gregory G. Finn. *Routing and Addressing Problems in Large Metropolitan-Scale Internetworks*. Technical report ISI/RR-87-180. University of Southern California, Marina del Rey, Information Sciences Institute, 1987.
- [FM07] Stefan Funke and Nikola Milosavljević. „Guaranteed-Delivery Geographic Routing Under Uncertain Node Locations.“ In: *Proc. 26th IEEE Internat. Conf. Computer Communications (INFOCOM'07)*. IEEE, 2007, 1244–1252. DOI: [10.1109/INFCOM.2007.148](https://doi.org/10.1109/INFCOM.2007.148).
- [Fon+05] Rodrigo Fonseca, Sylvia Ratnasamy, Jerry Zhao, Cheng Tien Ee, David Culler, Scott Shenker, and Ion Stoica. „Beacon Vector Routing: Scalable Point-to-Point Routing in Wireless Sensor Networks.“ In: *Proc. 2nd Usenix/ACM Symp. Networked Systems Design and Implementation (NSDI'05)*. USENIX Association, 2005, 329–342.
- [FPW09] Roland Flury, Sriram V. Pemmaraju, and Roger Wattenhofer. „Greedy Routing with Bounded Stretch.“ In: *Proc. 28th IEEE Internat. Conf. Computer Communications (INFOCOM'09)*. IEEE, 2009, 1737–1745. DOI: [10.1109/INFCOM.2009.5062093](https://doi.org/10.1109/INFCOM.2009.5062093).
- [FRS09] Hannes Frey, Stefan Rührup, and Ivan Stojmenović. „Routing in Wireless Sensor Networks.“ In: *Guide to Wireless Sensor Networks*. Ed. by Subhas Chandra Misra, Isaac Woungang, and Sudip Misra. *Computer Communications and Networks*. Springer, 2009, 81–111. DOI: [10.1007/978-1-84882-218-4_4](https://doi.org/10.1007/978-1-84882-218-4_4).
- [FS06] Hannes Frey and Ivan Stojmenović. „On Delivery Guarantees of Face and Combined Greedy-Face Routing in Ad Hoc and Sensor Networks.“ In: *Proc. 12th ACM Internat. Conf. Mobile Computing and Networking (MobiCom'06)*. ACM, 2006, 390–401. DOI: [10.1145/1161089.1161133](https://doi.org/10.1145/1161089.1161133).

- [Gao+05] Jie Gao, Leonidas J. Guibas, John Hershberger, Li Zhang, and An Zhu. „Geometric Spanners for Routing in Mobile Networks.“ In: *IEEE J. Selected Areas in Communications* 23.1 (2005), 174–185. DOI: [10.1109/JSAC.2004.837364](https://doi.org/10.1109/JSAC.2004.837364).
- [GG12] Jie Gao and Leonidas J. Guibas. „Geometric Algorithms for Sensor Networks.“ In: *Phil. Trans. R. Soc. A* 370.1958 (2012), 27–51. DOI: [10.1098/rsta.2011.0215](https://doi.org/10.1098/rsta.2011.0215).
- [GGL15] Jie Gao, Xianfeng David Gu, and Feng Luo. „Discrete Ricci Flow for Geometric Routing.“ In: *Encyclopedia of Algorithms*. Ed. by Ming-Yang Kao. Springer, 2015, 1–8. DOI: [10.1007/978-3-642-27848-8_602-1](https://doi.org/10.1007/978-3-642-27848-8_602-1).
- [GS09] Michael T. Goodrich and Darren Strash. „Succinct Greedy Geometric Routing in the Euclidean Plane.“ In: *Proc. 20th Internat. Symp. Algorithms and Computation (ISAAC'09)*. Ed. by Yingfei Dong, Ding-Zhu Du, and Oscar Ibarra. Vol. 5878 of *Lecture Notes Comput. Sci.* Springer, 2009, 781–791. DOI: [10.1007/978-3-642-10631-6_79](https://doi.org/10.1007/978-3-642-10631-6_79).
- [GUR7.5] *Gurobi Optimizer*. Version 7.5. Gurobi Optimization, Inc. 2017. URL: <http://www.gurobi.com>.
- [GVY97] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. „Primal-Dual Approximation Algorithms for Integral Flow and Multicut in Trees.“ In: *Algorithmica* 18.1 (1997), 3–20. DOI: [10.1007/BF02523685](https://doi.org/10.1007/BF02523685).
- [HE05] Weidong Huang and Peter Eades. „How People Read Graphs.“ In: *Proc. 4th Asia Pacific Symposium on Information Visualisation (APVIS'05)*. Australian Comput. Soc., 2005, 51–58.
- [HEH09] Weidong Huang, Peter Eades, and Seok-Hee Hong. „A Graph Reading Behavior: Geodesic-Path Tendency.“ In: *Proc. 2nd IEEE Pacific Visualization Symp. (PacificVis'09)*. IEEE, 2009, 137–144. DOI: [10.1109/PACIFICVIS.2009.4906848](https://doi.org/10.1109/PACIFICVIS.2009.4906848).
- [HH15] Xin He and Dayu He. „Monotone Drawings of 3-Connected Plane Graphs.“ In: *Proc. 23rd European Symp. Algorithms (ESA'15)*. Ed. by Nikhil Bansal and Irene Finocchi. Vol. 9294 of *Lecture Notes Comput. Sci.* Springer, 2015, 729–741. DOI: [10.1007/978-3-662-48350-3_61](https://doi.org/10.1007/978-3-662-48350-3_61).
- [HH17] Dayu He and Xin He. „Optimal Monotone Drawings of Trees.“ In: *SIAM J. Discrete Math.* (2017), 1867–1877. DOI: [10.1137/16M1080045](https://doi.org/10.1137/16M1080045).
- [HR15] Md. Iqbal Hossain and Md. Saidur Rahman. „Good Spanning Trees in Graph Drawing.“ In: *Theoret. Comput. Sci.* 607.2 (2015), 149–165. DOI: [10.1016/j.tcs.2015.09.004](https://doi.org/10.1016/j.tcs.2015.09.004).
- [Hua07] Weidong Huang. „Using Eye Tracking to Investigate Graph Layout Effects.“ In: *Proc. 6th Internat. Asia-Pacific Symp. Visualization (APVIS'07)*. IEEE, 2007, 97–100. DOI: [10.1109/APVIS.2007.329282](https://doi.org/10.1109/APVIS.2007.329282).

- [IKL99] Christian Icking, Rolf Klein, and Elmar Langetepe. „Self-Approaching Curves.“ In: *Math. Proc. Camb. Phil. Soc.* 125.3 (1999), 441–453. DOI: [10.1017/S0305004198003016](https://doi.org/10.1017/S0305004198003016).
- [Jia+15] Hongbo Jiang, Tianlong Yu, Chen Tian, Guang Tan, and Chonggang Wang. „Connectivity-Based Segmentation in Large-Scale 2-D/3-D Sensor Networks: Algorithm and Applications.“ In: *IEEE/ACM Trans. Networking* 23.1 (2015), 15–27. DOI: [10.1109/TNET.2013.2289912](https://doi.org/10.1109/TNET.2013.2289912).
- [JM96] David B. Johnson and David A. Maltz. „Dynamic Source Routing in Ad Hoc Wireless Networks.“ In: *Mobile Computing*. Ed. by Tomasz Imielinski and Henry F. Korth. Vol. 353 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, 1996, 153–181. DOI: [10.1007/978-0-585-29603-6_5](https://doi.org/10.1007/978-0-585-29603-6_5).
- [Kei00] J. Mark Keil. „Polygon Decomposition.“ In: *Handbook of Computational Geometry*. Ed. by Jörg-Rüdiger Sack and Jorge Urrutia. North-Holland, 2000, 491–518. DOI: [10.1016/B978-044482537-7/50012-7](https://doi.org/10.1016/B978-044482537-7/50012-7).
- [Kei85] J. Mark Keil. „Decomposing a Polygon into Simpler Components.“ In: *SIAM J. Comput.* 14.4 (1985), 799–817. DOI: [10.1137/0214056](https://doi.org/10.1137/0214056).
- [Kim+05a] Young-Jin Kim, Ramesh Govindan, Brad Karp, and Scott Shenker. „Geographic Routing Made Practical.“ In: *Proc. 2nd Usenix/ACM Symp. Networked Systems Design and Implementation (NSDI'05)*. USENIX Association, 2005, 217–230.
- [Kim+05b] Young-Jin Kim, Ramesh Govindan, Brad Karp, and Scott Shenker. „On the Pitfalls of Geographic Face Routing.“ In: *Proc. Joint Workshop on Foundations of Mobile Computing (DIALM-POMC'05)*. ACM, 2005, 34–43. DOI: [10.1145/1080810.1080818](https://doi.org/10.1145/1080810.1080818).
- [Kin+14] Philipp Kindermann, André Schulz, Joachim Spoerhase, and Alexander Wolff. „On Monotone Drawings of Trees.“ In: *Proc. 22nd Internat. Symp. Graph Drawing (GD'14)*. Ed. by Christian A. Duncan and Antonios Symvonis. Vol. 8871 of *Lecture Notes Comput. Sci.* Springer, 2014, 488–500. DOI: [10.1007/978-3-662-45803-7_41](https://doi.org/10.1007/978-3-662-45803-7_41).
- [KK00] Brad Karp and H. T. Kung. „GPSR: Greedy Perimeter Stateless Routing for Wireless Networks.“ In: *Proc. 6th ACM Internat. Conf. Mobile Computing and Networking (MobiCom'00)*. ACM, 2000, 243–254. DOI: [10.1145/345910.345953](https://doi.org/10.1145/345910.345953).
- [Kle07] Robert Kleinberg. „Geographic Routing Using Hyperbolic Space.“ In: *Proc. 26th IEEE Internat. Conf. Computer Communications (INFOCOM'07)*. IEEE, 2007, 1902–1909. DOI: [10.1109/INFCOM.2007.221](https://doi.org/10.1109/INFCOM.2007.221).
- [KNR92] Sampath Kannan, Moni Naor, and Steven Rudich. „Implicit Representation of Graphs.“ In: *SIAM J. Discrete Math.* 5.4 (1992), 596–603. DOI: [10.1137/0405049](https://doi.org/10.1137/0405049).

- [Kob12] Stephen G. Kobourov. *Spring Embedders and Force Directed Graph Drawing Algorithms*. 2012. arXiv: [1201.3011](https://arxiv.org/abs/1201.3011).
- [KR92] Donald E. Knuth and Arvind Raghunathan. „The Problem of Compatible Representatives.“ In: *SIAM J. Discrete Math.* 5.3 (1992), 422–427. DOI: [10.1137/0405033](https://doi.org/10.1137/0405033).
- [KSU99] Evangelos Kranakis, Harvinder Singh, and Jorge Urrutia. „Compass Routing on Geometric Networks.“ In: *Proc. 11th Canadian Conf. Computational Geometry (CCCG'99)*. 1999, 51–54.
- [KW05] Holger Karl and Andreas Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2005. DOI: [10.1002/0470095121](https://doi.org/10.1002/0470095121).
- [KWZo8a] Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. „Ad Hoc Networks Beyond Unit Disk Graphs.“ In: *Wireless Networks* 14.5 (2008), 715–729. DOI: [10.1007/s11276-007-0045-6](https://doi.org/10.1007/s11276-007-0045-6).
- [KWZo8b] Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. „An Algorithmic Approach to Geographic Routing in Ad Hoc and Sensor Networks.“ In: *IEEE/ACM Trans. Networking* 16.1 (2008), 51–62. DOI: [10.1109/TNET.2007.900372](https://doi.org/10.1109/TNET.2007.900372).
- [LA06] Ke Liu and Nael Abu-Ghazaleh. „Aligned Virtual Coordinates for Greedy Routing in WSNs.“ In: *Proc. 3rd IEEE Internat. Conf. Mobile Ad-hoc and Sensor Systems (MASS'06)*. IEEE, 2006, 377–386. DOI: [10.1109/MOBHOC.2006.278577](https://doi.org/10.1109/MOBHOC.2006.278577).
- [Lee+06] Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry. „Task Taxonomy for Graph Visualization.“ In: *Proc. AVI Workshop on BEYOND Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV'06)*. ACM, 2006, 1–5. DOI: [10.1145/1168149.1168168](https://doi.org/10.1145/1168149.1168168).
- [Li+00] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris. „A Scalable Location Service for Geographic Ad Hoc Routing.“ In: *Proc. 6th ACM Internat. Conf. Mobile Computing and Networking (MobiCom'00)*. ACM, 2000, 120–130. DOI: [10.1145/345910.345931](https://doi.org/10.1145/345910.345931).
- [Li+05] Chih-ping Li, Wei-jen Hsu, Bhaskar Krishnamachari, and Ahmed Helmy. „A Local Metric for Geographic Routing with Power Control in Wireless Networks.“ In: *Proc. 2nd Ann. IEEE Communications Society Conf. Sensor and Ad Hoc Communications and Networks (SECON'05)*. IEEE, 2005, 229–239. DOI: [10.1109/SAHNCN.2005.1557078](https://doi.org/10.1109/SAHNCN.2005.1557078).
- [Lic82] David Lichtenstein. „Planar Formulae and Their Uses.“ In: *SIAM J. Comput.* 11.2 (1982), 329–343. DOI: [10.1137/0211025](https://doi.org/10.1137/0211025).
- [Lin82] Andrzej Lingas. „The Power of Non-Rectilinear Holes.“ In: *Proc. 9th Internat. Colloquium on Automata, Languages, and Programming (ICALP'82)*. Ed. by Erik Meineche Schmidt and Mogens Nielsen. Vol. 140 of *Lecture Notes Comput. Sci.* Springer, 1982, 369–383. DOI: [10.1007/BFb0012784](https://doi.org/10.1007/BFb0012784).

- [LLM06] Ben Leong, Barbara Liskov, and Robert Morris. „Geographic Routing Without Planarization.“ In: *Proc. 3rd Usenix/ACM Symp. Networked Systems Design and Implementation (NSDI'06)*. USENIX Association, 2006, 339–352.
- [LLM07] Ben Leong, Barbara Liskov, and Robert Morris. „Greedy Virtual Coordinates for Geographic Routing.“ In: *Proc. 15th IEEE Internat. Conf. Network Protocols (ICNP'07)*. IEEE, 2007, 71–80. DOI: [10.1109/ICNP.2007.4375838](https://doi.org/10.1109/ICNP.2007.4375838).
- [LLW88] Nati Linial, Laszlo Lovász, and Avi Wigderson. „Rubber Bands, Convex Embeddings and Graph Connectivity.“ In: *Combinatorica* 8.1 (1988), 91–102. DOI: [10.1007/BF02122557](https://doi.org/10.1007/BF02122557).
- [LM10] Tom Leighton and Ankur Moitra. „Some Results on Greedy Embeddings in Metric Spaces.“ In: *Discrete Comput. Geom.* 44.3 (2010), 686–705. DOI: [10.1007/s00454-009-9227-6](https://doi.org/10.1007/s00454-009-9227-6).
- [LO17] Anna Lubiw and Joseph O'Rourke. „Angle-Monotone Paths in Non-Obtuse Triangulations.“ In: *Proc. 29th Canadian Conf. Computational Geometry (CCCG'17)*. 2017, 25–30. arXiv: [1707.00219](https://arxiv.org/abs/1707.00219).
- [MLo8] Ankur Moitra and Tom Leighton. „Some Results on Greedy Embeddings in Metric Spaces.“ Presentation given at the 49th Ann. IEEE Symp. Foundations of Computer Science (FOCS'08). 2008. URL: <http://people.csail.mit.edu/moitra/docs/ftl.pdf>.
- [MS15] Konstantinos Mastakas and Antonios Symvonis. „On the Construction of Increasing-Chord Graphs on Convex Point Sets.“ In: *Proc. 6th Internat. Conf. Information, Intelligence, Systems and Applications (IISA'15)*. IEEE, 2015, 1–6. DOI: [10.1109/IISA.2015.7388028](https://doi.org/10.1109/IISA.2015.7388028).
- [Ngu+07] An Nguyen, Nikola Milosavljević, Qing Fang, Jie Gao, and Leonidas J. Guibas. „Landmark Selection and Greedy Landmark-Descent Routing for Sensor Networks.“ In: *Proc. 26th IEEE Internat. Conf. Computer Communications (INFOCOM'07)*. IEEE, 2007, 661–669. DOI: [10.1109/INFCOM.2007.83](https://doi.org/10.1109/INFCOM.2007.83).
- [NP13] Martin Nöllenburg and Roman Prutkin. „Euclidean Greedy Drawings of Trees.“ In: *Proc. 21st European Symp. Algorithms (ESA'13)*. Ed. by Hans L. Bodlaender and Giuseppe F. Italiano. Vol. 8125 of *Lecture Notes Comput. Sci.* Springer, 2013, 767–778. DOI: [10.1007/978-3-642-40450-4_65](https://doi.org/10.1007/978-3-642-40450-4_65).
- [NP17] Martin Nöllenburg and Roman Prutkin. „Euclidean Greedy Drawings of Trees.“ In: *Discrete Comput. Geom.* 58.3 (2017), 543–579. DOI: [10.1007/s00454-017-9913-8](https://doi.org/10.1007/s00454-017-9913-8).
- [NPR14] Martin Nöllenburg, Roman Prutkin, and Ignaz Rutter. „On Self-Approaching and Increasing-Chord Drawings of 3-Connected Planar Graphs.“ In: *Proc. 22nd Internat. Symp. Graph Drawing (GD'14)*. Ed. by Christian A. Duncan and Antonios Symvonis. Vol. 8871 of *Lecture Notes Comput. Sci.* Springer, 2014, 476–487. DOI: [10.1007/978-3-662-45803-7_40](https://doi.org/10.1007/978-3-662-45803-7_40).

- [NPR15] Martin Nöllenburg, Roman Prutkin, and Ignaz Rutter. „Partitioning Graph Drawings and Triangulated Simple Polygons into Greedily Routable Regions.“ In: *Proc. 26th Internat. Symp. Algorithms and Computation (ISAAC'15)*. Ed. by Khaled Elbassioni and Kazuhisa Makino. Vol. 9472 of *Lecture Notes Comput. Sci.* Springer, 2015, 637–649. DOI: [10.1007/978-3-662-48971-0_54](https://doi.org/10.1007/978-3-662-48971-0_54).
- [NPR16] Martin Nöllenburg, Roman Prutkin, and Ignaz Rutter. „On Self-Approaching and Increasing-Chord Drawings of 3-Connected Planar Graphs.“ In: *J. Comput. Geom.* 7.1 (2016), 47–69. DOI: [10.20382/jocg.v7i1a3](https://doi.org/10.20382/jocg.v7i1a3).
- [NPR17] Martin Nöllenburg, Roman Prutkin, and Ignaz Rutter. „Partitioning Graph Drawings and Triangulated Simple Polygons into Greedily Routable Regions.“ In: *Internat. J. Comput. Geom. Appl.* 27 (1–2 2017). Special issue on selected papers from ISAAC'15, 121–158. DOI: [10.1142/S0218195917600068](https://doi.org/10.1142/S0218195917600068).
- [NS03] James Newsome and Dawn Song. „GEM: Graph EMbedding for Routing and Data-Centric Storage in Sensor Networks Without Geographic Information.“ In: *Proc. 1st ACM Internat. Conf. Embedded Networked Sensor Systems (SenSys'03)*. ACM, 2003, 76–88. DOI: [10.1145/958491.958501](https://doi.org/10.1145/958491.958501).
- [NS07] Giri Narasimhan and Michiel Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007. DOI: [10.1017/CB09780511546884](https://doi.org/10.1017/CB09780511546884).
- [OS17] Anargyros Oikonomou and Antonios Symvonis. „Simple Compact Monotone Tree Drawings.“ In: *Proc. 25th Internat. Symp. Graph Drawing and Network Visualization (GD'17)*. Ed. by Fabrizio Frati and Kwan-Liu Ma. Vol. 10692 of *Lecture Notes Comput. Sci.* Springer, 2017, 326–333. DOI: [10.1007/978-3-319-73915-1_26](https://doi.org/10.1007/978-3-319-73915-1_26).
- [PB94] Charles E. Perkins and Pravin Bhagwat. „Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers.“ In: *Proc. ACM Conf. Communications Architectures, Protocols and Applications (SIGCOMM'94)*. ACM, 1994, 234–244. DOI: [10.1145/190314.190336](https://doi.org/10.1145/190314.190336).
- [PNV13] Nikolaos A. Pantazis, Stefanos A. Nikolidakis, and Dimitrios D. Vergados. „Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey.“ In: *IEEE Communications Surveys and Tutorials* 15.2 (2013), 551–591. DOI: [10.1109/SURV.2012.062612.00084](https://doi.org/10.1109/SURV.2012.062612.00084).
- [PR05] Christos H. Papadimitriou and David Ratajczak. „On a Conjecture Related to Geometric Routing.“ In: *Theoret. Comput. Sci.* 344.1 (2005), 3–14. DOI: [10.1016/j.tcs.2005.06.022](https://doi.org/10.1016/j.tcs.2005.06.022).
- [PR99] Charles E. Perkins and Elizabeth M. Royer. „Ad-Hoc On-Demand Distance Vector Routing.“ In: *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*. IEEE, 1999, 90–100. DOI: [10.1109/MCSA.1999.749281](https://doi.org/10.1109/MCSA.1999.749281).

- [Pur+12] Helen C. Purchase, John Hamer, Martin Nöllenburg, and Stephen G. Kobourov. „On the Usability of Lombardi Graph Drawings.“ In: *Proc. 20th Internat. Symp. Graph Drawing (GD'12)*. Ed. by Walter Didimo and Maurizio Patrignani. Vol. 7704 of *Lecture Notes Comput. Sci.* Springer, 2012, 451–462. DOI: [10.1007/978-3-642-36763-2_40](https://doi.org/10.1007/978-3-642-36763-2_40).
- [Rao+03] Ananth Rao, Sylvia Ratnasamy, Christos H. Papadimitriou, Scott Shenker, and Ion Stoica. „Geographic Routing without Location Information.“ In: *Proc. 9th ACM Internat. Conf. Mobile Computing and Networking (MobiCom'03)*. ACM, 2003, 96–108. DOI: [10.1145/938985.938996](https://doi.org/10.1145/938985.938996).
- [Raw+14] Priyanka Rawat, Kamal Deep Singh, Hakima Chaouchi, and Jean Marie Bonnin. „Wireless Sensor Networks: A Survey on Recent Developments and Potential Synergies.“ In: *J. Supercomput.* 68.1 (2014), 1–48. DOI: [10.1007/s11227-013-1021-9](https://doi.org/10.1007/s11227-013-1021-9).
- [RFC3626] Thomas H. Clausen and Philippe Jacquet. *Optimized Link State Routing Protocol (OLSR)*. RFC 3626. RFC Editor, 2003. DOI: [10.17487/rfc3626](https://doi.org/10.17487/rfc3626).
- [Rot94] Günter Rote. „Curves with Increasing Chords.“ In: *Math. Proc. Camb. Phil. Soc.* 115.1 (1994), 1–12. DOI: [10.1017/S0305004100071875](https://doi.org/10.1017/S0305004100071875).
- [Sar+09] Rik Sarkar, Xiaotian Yin, Jie Gao, Feng Luo, and Xianfeng David Gu. „Greedy Routing with Guaranteed Delivery Using Ricci Flows.“ In: *Proc. 8th ACM/IEEE Internat. Conf. Information Processing in Sensor Networks (IPSN'09)*. IEEE, 2009, 121–132.
- [Sar+10] Rik Sarkar, Wei Zeng, Jie Gao, and Xianfeng David Gu. „Covering Space for In-Network Sensor Data Storage.“ In: *Proc. 9th ACM/IEEE Internat. Conf. Information Processing in Sensor Networks (IPSN'10)*. ACM, 2010, 232–243. DOI: [10.1145/1791212.1791240](https://doi.org/10.1145/1791212.1791240).
- [Sch90] Walter Schnyder. „Embedding Planar Graphs on the Grid.“ In: *Proc. First ACM-SIAM Symp. Discrete Algorithms (SODA'90)*. SIAM, 1990, 138–148.
- [Sea+04] Karim Seada, Marco Zuniga, Ahmed Helmy, and Bhaskar Krishnamachari. „Energy-Efficient Forwarding Strategies for Geographic Routing in Lossy Wireless Sensor Networks.“ In: *Proc. 2nd ACM Internat. Conf. Embedded Networked Sensor Systems (SenSys'04)*. ACM, 2004, 108–121. DOI: [10.1145/1031495.1031509](https://doi.org/10.1145/1031495.1031509).
- [SL01] Ivan Stojmenović and Xu Lin. „Loop-Free Hybrid Single-Path/Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks.“ In: *IEEE Trans. Parallel and Distributed Systems* 12.10 (2001), 1023–1032. DOI: [10.1109/71.963415](https://doi.org/10.1109/71.963415).
- [SSGo7] Sundar Subramanian, Sanjay Shakkottai, and Piyush Gupta. „On Optimal Geographic Routing in Wireless Networks with Holes and Non-Uniform Traffic.“ In: *Proc. 26th IEEE Internat. Conf. Computer Communications (INFOCOM'07)*. IEEE, 2007, 1019–1027. DOI: [10.1109/INFCOM.2007.123](https://doi.org/10.1109/INFCOM.2007.123).

- [SVZ07] Christian Schindelhauer, Klaus Volbert, and Martin Ziegler. „Geometric Spanners with Applications in Wireless Networks.“ In: *Comput. Geom. Theory Appl.* 36.3 (2007), 197–214. DOI: [10.1016/j.comgeo.2006.02.001](https://doi.org/10.1016/j.comgeo.2006.02.001).
- [TBK09] Guang Tan, Marin Bertier, and Anne-Marie Kermarrec. „Convex Partition of Sensor Networks and Its Use in Virtual Coordinate Geographic Routing.“ In: *Proc. 28th IEEE Internat. Conf. Computer Communications (INFOCOM'09)*. IEEE, 2009, 1746–1754. DOI: [10.1109/INFCOM.2009.5062094](https://doi.org/10.1109/INFCOM.2009.5062094).
- [Thu85] William P. Thurston. „The Finite Riemann Mapping Theorem.“ Invited talk at the International Symposium at Purdue University on the Occasion of the Proof of the Bieberbach Conjecture. 1985.
- [TK12] Guang Tan and Anne-Marie Kermarrec. „Greedy Geographic Routing in Large-Scale Sensor Networks: A Minimum Network Decomposition Approach.“ In: *IEEE/ACM Trans. Networking* 20.3 (2012), 864–877. DOI: [10.1109/TNET.2011.2167758](https://doi.org/10.1109/TNET.2011.2167758).
- [TY94] Robin Thomas and Xingxing Yu. „4-Connected Projective-Planar Graphs Are Hamiltonian.“ In: *J. Combinatorial Theory, Series B* 62.1 (1994), 114–132. DOI: [10.1006/jctb.1994.1058](https://doi.org/10.1006/jctb.1994.1058).
- [TZ01] Mikkel Thorup and Uri Zwick. „Compact Routing Schemes.“ In: *Proc. 13th Ann. ACM Symp. Parallel Algorithms and Architectures (SPAA'01)*. ACM, 2001, 1–10. DOI: [10.1145/378580.378581](https://doi.org/10.1145/378580.378581).
- [War+02] Colin Ware, Helen C. Purchase, Linda Colpoys, and Matthew McGill. „Cognitive Measurements of Graph Aesthetics.“ In: *Information Visualization* 1.2 (2002), 103–110. DOI: [10.1057/palgrave.ivs.9500013](https://doi.org/10.1057/palgrave.ivs.9500013).
- [Wat+09] Thomas Watteyne, Isabelle Augé-Blum, Mischa Dohler, Stéphane Ubéda, and Dominique Barthel. „Centroid Virtual Coordinates – A Novel Near-Shortest Path Routing Paradigm.“ In: *Computer Networks* 53.10 (2009), 1697–1711. DOI: [10.1016/j.comnet.2008.12.017](https://doi.org/10.1016/j.comnet.2008.12.017).
- [WH14] Jiun-Jie Wang and Xin He. „Succinct Strictly Convex Greedy Drawing of 3-Connected Plane Graphs.“ In: *Theoret. Comput. Sci.* 532 (2014), 80–90. DOI: [10.1016/j.tcs.2013.05.024](https://doi.org/10.1016/j.tcs.2013.05.024).
- [WW07] Dorothea Wagner and Roger Wattenhofer, eds. *Algorithms for Sensor and Ad Hoc Networks*. Vol. 4621 of *Lecture Notes Comput. Sci.* Springer, 2007. DOI: [10.1007/978-3-540-74991-2](https://doi.org/10.1007/978-3-540-74991-2).
- [XWJ14] Su Xia, Hongyi Wu, and Miao Jin. „GPS-Free Greedy Routing With Delivery Guarantee and Low Stretch Factor on 2-D and 3-D Surfaces.“ In: *IEEE Internet of Things J.* 1.3 (2014), 233–242. DOI: [10.1109/JIOT.2014.2320260](https://doi.org/10.1109/JIOT.2014.2320260).
- [Yvi17] Mariette Yvinec. *2D Triangulation*. CGAL User and Reference Manual. Version 4.10. CGAL Editorial Board. 2017. URL: <http://doc.cgal.org/4.10/Manual/packages.html#PkgTriangulation2Summary>.

- [Zen+07] Kai Zeng, Wenjing Lou, Jie Yang, and Donald R. Brown. „On Throughput Efficiency of Geographic Opportunistic Routing in Multihop Wireless Networks.“ In: *Mobile Netw. Appl.* 12 (5–6 2007), 347–357. DOI: [10.1007/s11036-008-0051-7](https://doi.org/10.1007/s11036-008-0051-7).
- [ZG13] Huaming Zhang and Swetha Govindaiah. „Greedy Routing via Embedding Graphs onto Semi-Metric Spaces.“ In: *Theoret. Comput. Sci.* 508 (2013), 26–34. DOI: [10.1016/j.tcs.2012.01.049](https://doi.org/10.1016/j.tcs.2012.01.049).
- [Zho+10] Jiangwei Zhou, Yu Chen, Ben Leong, and Pratibha Sundar Sundaramoorthy. „Practical 3D Geographic Routing for Wireless Sensor Networks.“ In: *Proc. 8th ACM Internat. Conf. Embedded Networked Sensor Systems (SenSys'10)*. ACM, 2010, 337–350. DOI: [10.1145/1869983.1870016](https://doi.org/10.1145/1869983.1870016).
- [ZJ09] Jun Zheng and Abbas Jamalipour, eds. *Wireless Sensor Networks: A Networking Perspective*. Wiley-IEEE Press, 2009. DOI: [10.1002/9780470443521](https://doi.org/10.1002/9780470443521).
- [Zolo7] Aaron Zollinger. „Geographic Routing.“ In: *Algorithms for Sensor and Ad Hoc Networks*. Ed. by Dorothea Wagner and Roger Wattenhofer. Vol. 4621 of *Lecture Notes Comput. Sci.* Springer, 2007, 161–185. DOI: [10.1007/978-3-540-74991-2_9](https://doi.org/10.1007/978-3-540-74991-2_9).
- [ZR03] Michele Zorzi and Ramesh R. Rao. „Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance.“ In: *IEEE Trans. Mobile Computing* 2.4 (2003), 337–348. DOI: [10.1109/TMC.2003.1255648](https://doi.org/10.1109/TMC.2003.1255648).
- [ZSG07] Xianjin Zhu, Rik Sarkar, and Jie Gao. „Shape Segmentation and Applications in Sensor Networks.“ In: *Proc. 26th IEEE Internat. Conf. Computer Communications (INFOCOM'07)*. IEEE, 2007, 1838–1846. DOI: [10.1109/INFOCOM.2007.214](https://doi.org/10.1109/INFOCOM.2007.214).
- [ZSG09] Xianjin Zhu, Rik Sarkar, and Jie Gao. „Segmenting a Sensor Field: Algorithms and Applications in Network Design.“ In: *ACM Trans. Sensor Networks* 5.2 (2009), 12:1–12:32. DOI: [10.1145/1498915.1498918](https://doi.org/10.1145/1498915.1498918).

LIST OF PUBLICATIONS¹

ARTICLES IN PEER-REVIEWED JOURNALS

- [NP17] Martin Nöllenburg and Roman Prutkin. „Euclidean Greedy Drawings of Trees.“ In: *Discrete Comput. Geom.* 58.3 (2017), 543–579. DOI: [10.1007/s00454-017-9913-8](https://doi.org/10.1007/s00454-017-9913-8).
- [NPR17] Martin Nöllenburg, Roman Prutkin, and Ignaz Rutter. „Partitioning Graph Drawings and Triangulated Simple Polygons into Greedily Routable Regions.“ In: *Internat. J. Comput. Geom. Appl.* 27 (1–2 2017). Special issue on selected papers from ISAAC’15, 121–158. DOI: [10.1142/S0218195917600068](https://doi.org/10.1142/S0218195917600068).
- [NPR16] Martin Nöllenburg, Roman Prutkin, and Ignaz Rutter. „On Self-Approaching and Increasing-Chord Drawings of 3-Connected Planar Graphs.“ In: *J. Comput. Geom.* 7.1 (2016), 47–69. DOI: [10.20382/jocg.v7i1a3](https://doi.org/10.20382/jocg.v7i1a3).
- [NPR13] Martin Nöllenburg, Roman Prutkin, and Ignaz Rutter. „Edge-Weighted Contact Representations of Planar Graphs.“ In: *J. Graph Algorithms Appl.* 17.4 (2013). Special issue on selected papers from GD’12, 441–473. DOI: [10.7155/jgaa.00299](https://doi.org/10.7155/jgaa.00299).

ARTICLES IN PEER-REVIEWED CONFERENCE PROCEEDINGS

- [FP15] Fabian Fuchs and Roman Prutkin. „Simple Distributed $\Delta + 1$ Coloring in the SINR Model.“ In: *Proc. 22nd Internat. Colloquium on Structural Information and Communication Complexity (SIROCCO’15)*. Ed. by Christian Scheideler. Vol. 9439 of *Lecture Notes Comput. Sci.* Springer, 2015, 149–163. DOI: [10.1007/978-3-319-25258-2_11](https://doi.org/10.1007/978-3-319-25258-2_11).
- [KNP15a] Boris Klemz, Martin Nöllenburg, and Roman Prutkin. „Recognizing Weighted Disk Contact Graphs.“ In: *Proc. 23rd Internat. Symp. Graph Drawing and Network Visualization (GD’15)*. Ed. by Emilio Di Giacomo and Anna Lubiw. Vol. 9411 of *Lecture Notes Comput. Sci.* Springer, 2015, 433–446. DOI: [10.1007/978-3-319-27261-0_36](https://doi.org/10.1007/978-3-319-27261-0_36).
- [Nac+15] Lev Nachmanson, Roman Prutkin, Bongshin Lee, Nathalie Henry Riche, Alexander E. Holroyd, and Xiaoji Chen. „GraphMaps: Browsing Large Graphs as Interactive Maps.“ In: *Proc. 23rd Internat. Symp. Graph Drawing and Network Visualization (GD’15)*. Ed. by Emilio Di

¹ In theoretical computer science, the authors are usually ordered alphabetically.

- Giacomo and Anna Lubiw. Vol. 9411 of *Lecture Notes Comput. Sci.* Springer, 2015, 3–15. DOI: [10.1007/978-3-319-27261-0_1](https://doi.org/10.1007/978-3-319-27261-0_1).
- [NPR15] Martin Nöllenburg, Roman Prutkin, and Ignaz Rutter. „Partitioning Graph Drawings and Triangulated Simple Polygons into Greedily Routable Regions.“ In: *Proc. 26th Internat. Symp. Algorithms and Computation (ISAAC’15)*. Ed. by Khaled Elbassioni and Kazuhisa Makino. Vol. 9472 of *Lecture Notes Comput. Sci.* Springer, 2015, 637–649. DOI: [10.1007/978-3-662-48971-0_54](https://doi.org/10.1007/978-3-662-48971-0_54).
- [vLMP15] Moritz von Looz, Henning Meyerhenke, and Roman Prutkin. „Generating Random Hyperbolic Graphs in Subquadratic Time.“ In: *Proc. 26th Internat. Symp. Algorithms and Computation (ISAAC’15)*. Ed. by Khaled Elbassioni and Kazuhisa Makino. Vol. 9472 of *Lecture Notes Comput. Sci.* Springer, 2015, 467–478. DOI: [10.1007/978-3-662-48971-0_40](https://doi.org/10.1007/978-3-662-48971-0_40).
- [NPR14] Martin Nöllenburg, Roman Prutkin, and Ignaz Rutter. „On Self-Approaching and Increasing-Chord Drawings of 3-Connected Planar Graphs.“ In: *Proc. 22nd Internat. Symp. Graph Drawing (GD’14)*. Ed. by Christian A. Duncan and Antonios Symvonis. Vol. 8871 of *Lecture Notes Comput. Sci.* Springer, 2014, 476–487. DOI: [10.1007/978-3-662-45803-7_40](https://doi.org/10.1007/978-3-662-45803-7_40).
- [Bie+13] Therese Biedl, Thomas Bläsius, Benjamin Niedermann, Martin Nöllenburg, Roman Prutkin, and Ignaz Rutter. „Using ILP/SAT to Determine Pathwidth, Visibility Representations, and Other Grid-Based Graph Drawings.“ In: *Proc. 21st Internat. Symp. Graph Drawing (GD’13)*. Ed. by Stephen Wismath and Alexander Wolff. Vol. 8242 of *Lecture Notes Comput. Sci.* Springer, 2013, 460–471. DOI: [10.1007/978-3-319-03841-4_40](https://doi.org/10.1007/978-3-319-03841-4_40).
- [NP13] Martin Nöllenburg and Roman Prutkin. „Euclidean Greedy Drawings of Trees.“ In: *Proc. 21st European Symp. Algorithms (ESA’13)*. Ed. by Hans L. Bodlaender and Giuseppe F. Italiano. Vol. 8125 of *Lecture Notes Comput. Sci.* Springer, 2013, 767–778. DOI: [10.1007/978-3-642-40450-4_65](https://doi.org/10.1007/978-3-642-40450-4_65).
- [NPR12] Martin Nöllenburg, Roman Prutkin, and Ignaz Rutter. „Edge-Weighted Contact Representations of Planar Graphs.“ In: *Proc. 20th Internat. Symp. Graph Drawing (GD’12)*. Ed. by Walter Didimo and Maurizio Patrignani. Vol. 7704 of *Lecture Notes Comput. Sci.* Springer, 2012, 224–235. DOI: [10.1007/978-3-642-36763-2_20](https://doi.org/10.1007/978-3-642-36763-2_20).
- [PKD12] Roman Prutkin, Anton Kaplanyan, and Carsten Dachsbacher. „Reflective Shadow Map Clustering for Real-Time Global Illumination.“ In: *Proc. 33rd Ann. Conf. European Association for Computer Graphics (EuroGraphics’12), short papers*. Ed. by Carlos Andujar and Enrico

Puppo. The Eurographics Association, 2012, 9–12. DOI: [10 . 2312 / conf/EG2012/short/009-012](https://doi.org/10.2312/conf/EG2012/short/009-012).

OTHER PUBLICATIONS

- [Ang+18] Patrizio Angelini, Michael A. Bekos, Walter Didimo, Luca Grilli, Philipp Kindermann, Tamara Mchedlidze, Roman Prutkin, Antonios Symvonis, and Alessandra Tappini. „Greedy Rectilinear Orthogonal Drawings.“ Under submission. 2018.
- [Pru16] Roman Prutkin. *A Note on the Area Requirement of Euclidean Greedy Embeddings of Christmas Cactus Graphs*. 2016. arXiv: [1605.09244](https://arxiv.org/abs/1605.09244).
- [KNP15b] Boris Klemz, Martin Nöllenburg, and Roman Prutkin. „Recognizing Weighted Disk Contact Graphs.“ In: *Proc. 31st European Workshop on Computational Geometry (EuroCG'15)*. 2015, 113–116.

SOFTWARE

- [NP15] Lev Nachmanson and Roman Prutkin. *GraphMaps: Browsing Large Graphs as Interactive Maps*. In collab. with Sergey Pupyrev, Tim Dwyer, and Ted Hart. Part of the *Microsoft Automatic Graph Layout* library. 2015. URL: <https://github.com/Microsoft/automatic-graph-layout#graphmaps>.

DECLARATION

Ich versichere hiermit, diese Dissertation selbstständig angefertigt und alle benutzten Hilfsmittel vollständig angegeben zu haben. Außerdem versichere ich, kenntlich gemacht zu haben, was aus Arbeiten anderer und eigener Veröffentlichungen unverändert oder mit Änderungen entnommen wurde.

Karlsruhe, 21. Februar 2018

Roman Prutkin